NIST Special Publication 1239-2

NIST Conference Papers
Fiscal Year 2017
Volume 2:
Communications Technology Laboratory
Information Technology Laboratory
Material Measurement Laboratory

Compiled and edited by:
Andrea Medina-Smith
Kathryn Miller
Karen Wick

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1239-2

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Special Publication 1239-2

NIST Conference Papers
Fiscal Year 2017
Volume 2:
Communications Technology Laboratory
Information Technology Laboratory
Material Measurement Laboratory

Compiled and edited by:
Andrea Medina-Smith
Kathryn Miller
Karen Wick
Information Services Office

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1239-2

August 2019

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 1239-2
Natl. Inst. Stand. Technol. Spec. Publ. 1239-2, 474 pages (August 2019)
CODEN: NSPUE2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Foreword

NIST is committed to the idea that results of federally funded research are a valuable national resource and a strategic asset. To the extent feasible and consistent with law, agency mission, resource constraints, and U.S. national, homeland, and economic security, NIST will promote the deposit of scientific data arising from unclassified research and programs, funded wholly or in part by NIST, except for Standard Reference Data, free of charge in publicly accessible databases. Subject to the same conditions and constraints listed above, NIST also intends to make freely available to the public, in publicly accessible repositories, all peer-reviewed scholarly publications arising from unclassified research and programs funded wholly or in part by NIST.

This Special Publication represents the work of Communications Technology Laboratory, Information Technology Laboratory, and Material Measurement Laboratory researchers at professional conferences, as reported in Fiscal Year 2017.

More information on public access to NIST research is available at https://www.nist.gov/open**.**

Table of Contents

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Communications Technology Laboratory

Work of researchers at professional conferences as reported in Fiscal Year 2017

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Variability of Sounder Measurements in Manufacturing Facilities

J. Quimby, A.E. Curtin, D. R. Novotny, K.A. Remley
NIST Communications Technology Laboratory
Boulder, CO, USA
Jeanne.quimby@nist.gov
David.novotny@nist.gov
Kate.Remley@nist.gov

C.-M. Wang
NIST Information Technology Laboratory
Boulder, CO, USA
Chih-ming.wang@nist.gov

R. Candell
NIST Engineering Laboratory
Gaithersburg, MD, USA
Richard.candell@nist.gov

***Abstract*—Uncertainties in the linear regression fit of Path Loss are derived from variability in position. These uncertainties are used to determine the required positional accuracy for channel sounder measurements.**

Keywords—Channel Sounders, Microwave Measurement, Uncertainty Analysis, Wireless System

I. Introduction

The National Institute of Standards and Technology (NIST) is characterizing manufacturing environments through measurements of the channel impulse response of representative production facilities. Using a pseudo-noise channel probe signal [1], a band-limited noise-like signal is transmitted over the channel. Since the signal's autocorrelation function approximates an impulse, the channel impulse response (CIR) can be measured by correlating the received signal with the transmitted pseudo-noise code. This measured impulse response can be used to derive the path loss.

Relating measured path loss and other channel parameters to positioning a physical manufacturing environment can be done with differing levels of accuracy. A robot, which provides location information while measuring the channel, is highly accurate but costly. A human, pushing a cart during a measurement run, is more cost effective but less accurate. While cost is an important parameter, accuracy in physical location like check points is highly prized by system engineers deploying wireless networks using estimated path loss. Therefore, metrics that provide insight into the required accuracy between a single check point and path loss are useful when choosing between a robot or human walking approach. In this contribution, uncertainties are derived from a linear regression fit of measured path loss from statistical analysis.

II. Techincal Discussion

A. Variability in Path Loss Linear Regression

The linear regression used to find path loss provides channel information through the calculation of the slope. The slope of the linear regression provides insight into the type and path loss severity of the channel. The linear regression model with one independent variable for path loss is shown in (1):

$$\rho = A + Br_t, \tag{1}$$

where ρ is the path loss in dB, r_t is the check point true values in meters, and A and B are sampled least-square estimators; r_t is not observed directly. Instead, the distance between the channel sounder's transmitter and receiver, r, is observed. Equation (1) becomes

$$\rho = A + Br, \tag{2}$$

$$r = r_t + r_{PE}, \tag{3}$$

where r_{PE} is the range position error which includes random perturbations. We assume that r_{PE} is distributed as a normal random variable with mean zero and a standard deviation of $\sigma_{r_{PE}}$. The sampled least-square estimators, A and B, for the linear regression model can be determined with a known $\sigma_{r_{PE}}$ [2].

B. Range Position Error, r_{PE}

We simulated a range position error using a Monte Carlo simulation for small x_{PE} and y_{PE} surrounding a single r_t. We use the standard deviation $\sigma_{r_{PE}}$ of the normal distribution to quantify the error in r_{PE} for different check point true values r_t in a factory environment. A variety of standard deviation values were calculated with a Monte Carlo Position Uncertainty algorithm. An example of this algorithm for perturbation in Cartesian x_{PE} and y_{PE} is shown in (4), (5), and (6):

$$x_{PE} = \pm v \cdot t \cdot \varsigma + \frac{1}{2} a \cdot t^2 \cdot \varsigma, \tag{4}$$

$$y_{PE} = \pm v \cdot t \cdot \varsigma + \frac{1}{2} a \cdot t^2 \cdot \varsigma, \tag{5}$$

$$\varsigma \sim U(0,1). \tag{6}$$

Candell, Richard; Curtin, Alexandra; Novotny, David; Quimby, Jeanne; Wang, Chih-Ming.
"Variability of Sounder Measurements in Manufacturing Facilities."
Paper presented at Institute of Electrical and Electronics Engineers (IEEE) Antenna and Propagation Society (AP-S) conference, Fajardo, Puerto Rico. June 26, 2016 - July 1, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

where ς is a pseudorandom value from a uniform distribution with the open interval of (0,1), v is velocity, t is time, and a is acceleration.

The standard deviation $\sigma_{r_{PE}} = \sqrt{\sigma_{r_{PE}}^2}$ is found with (7):

$$\sigma_{r_{PE}}^2 = \frac{\bar{x}_{PE}^2\sigma_{x_{PE}}^2 + \bar{y}_{PE}^2\sigma_{y_{PE}}^2 + 2\bar{x}_{x_{PE}}\bar{y}_{y_{PE}}\sigma_{xy_{PE}}}{\bar{x}_{PE}^2 + \bar{y}_{PE}^2}, \quad (7)$$

where $\bar{x}_{PE}$ is the mean of the $x_{PE}, \sigma_{x_{PE}}^2$ is the variance of $x_{PE}, \bar{y}_{PE}$ is the mean of the $y_{PE}, \sigma_{y_{PE}}^2$ is the variance of y, and $\sigma_{xy_{PE}}$ is the covariance of x_{PE} and y_{PE}. Fig. 1 shows our simulated $\sigma_{r_{PE}}$ versus independent values of r_t for 500 Monte Carlo simulations per r_t with t = 3 s, v = 1.4 m/s, a = 0.75 m/s^2.

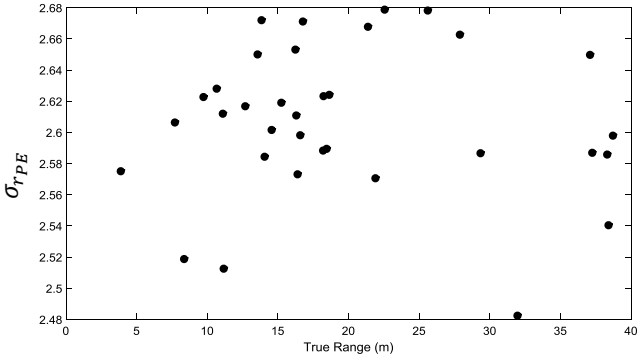


Fig. 1. Simulated Position Error for t = 3 s, v = 1.4 *m/s*, and a = 0.75 *m/s*^{2}.

Table 1 contains different simulated $\sigma_{r_{PE}}$ values based upon time, velocity, and acceleration.

TABLE 1: $\sigma_{r_{PE}}$ VERSUS TIME, VELOCITY, AND ACCELERATION

	t (s)	v (m/s)	a (m/s^2)	$\boldsymbol{\sigma_{r_{PE}}}$
Stationary	0	0	0	0
Robot	3	0.7	0.5	1.4
Human Walking	3	1.4	0.75	2.6
Human Running	5	10	1	29

C. Calculation of Uncertainity in A and B

With the assumption that the error in r_{PE} is quantified by $\sigma_{r_{PE}}$, B, and A and their uncertainties can be computed with (8) – (13):

$$B = \frac{\sigma_{\rho r}}{\sigma_r^2 - \sigma_{r_{PE}}^2}, \quad (8)$$

$$A = \bar{\rho} - B\bar{r}, \quad (9)$$

$$B_{Uncert} = \frac{1}{(M-1)\left(\sigma_r^2 - \sigma_{r_{PE}}^2\right)^2}\left(\sigma_r^2 s_{vv} + B^2\sigma_{r_{PE}}^2\right), \quad (10)$$

$$A_{Uncert} = (\bar{r})^2 B_{Uncert} + \frac{s_{vv}}{M}, \quad (11)$$

$$s_{vv} = \frac{1}{M-2}\sum_{m=1}^{M}[(\rho_m - \bar{\rho}) - (r_m - \bar{r})\,B]^2, \quad (12)$$

where $\sigma_{\rho r}$ is the covariance between the path loss ρ and the range r, σ_r^2 is the variance of the range, $\bar{\rho}$ is the mean of ρ, $\bar{r}$ is the mean of the range, and M is the number of samples.

In Table 2, the uncertainties for A and B were determined for different $\sigma_{r_{PE}}$. From Table 1, when $\sigma_{r_{PE}}$ equals 2.6 m, which corresponds to a human walking, the uncertainty of B was 0.054. Based upon these values, the channel sounder platform of a human walking lies within the two sigma bounds of the bounds of the Stationary case. Fig. 2 shows the standard path loss versus range with the uncertainties introduced by $\sigma_{r_{PE}}$ for a human walking as compared to path loss data collected for the Stationary case.

TABLE 2: B, and UNCERTAINTY B, A and UNCERTAINTY A

	B *(dB/m)*	Uncertainty B	A *(dB)*	Uncertainty A
Stationary	3	0.0036	21	0.62
Robot	4.9	0.026	-3	4.4
Human Walking	4.9	0.054	-3	9.2
Human Running	4.9	130	-3	2300

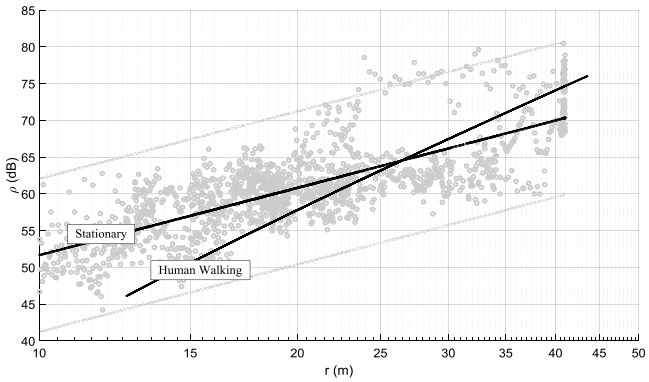


Fig. 2: Path loss versus range with uncertainties for a human walking.

III. CONCLUSION

B, A and their uncertainties provide insight into the required accuracy between physical location and path loss, when choosing between a robot or the human walking approach for different levels of accuracy in check \point position.

REFERENCES

[1] Papazian, P. B., Remley, K.A., Gentile, K. A, Golmie, N., "Radio channel sounders for modeling mobile commuications at 28 GHz, 60 GHz, and 83 GHz," IEEE Millimeter Waves, 2015 Global Symbposium, 2015, pg. 1-3

[2] Salous, Sana, *Radio Propagation Measurement and Channel Modeling*, Ed. Chichester, U. K., 2013

[3] W. Fuller, Measurement Error Models, John Wiley & Sons, inc, 1987

Publication of the United States government, not subject to copyright in the U.S.

Optimizing the UE Transmission Probability for D2D Direct Discovery

David Griffith and Fiona Lyons
National Institute of Standards and Technology
Gaithersburg, Maryland 20899–6730
Email: david.griffith@nist.gov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

***Abstract*—We model Mode 2 direct discovery in Device-to-Device (D2D) Long Term Evolution (LTE) networks and derive the optimal value of the discovery message transmission probability that minimizes the mean number of periods required for a successful discovery message transmission. We use Monte Carlo simulations to validate our analytical results and to show that optimizing the transmission probability produces nearly optimal performance with respect to the time required for all members of a group of User Equipments (UEs) to discover each other.**

I. Introduction

As an enhancement of Long Term Evolution (LTE) systems beyond Release 12, Device-to-Device (D2D) communications allow User Equipments (UEs) to exchange data directly with other D2D-capable UEs, rather than by routing data from one UE over an uplink connection to a base station, which forwards the data over a downlink to the destination UE. The D2D connection from one UE to another UE, defined as a *sidelink* (SL) as an extension of the uplink/downlink (UL/DL) nomenclature, consists of a set of resources that can be used in Time Division Duplex (TDD) or Frequency Division Duplex (FDD) LTE deployments [1].

D2D was originally proposed as an underlay network that would offload intracell traffic from base stations while minimally interfering with intercell traffic, thus increasing network throughput by as much as 65 % [2], but the concept has since expanded. D2D-capable devices can support cell coverage extension by serving as relays for UEs that are outside the coverage area of any cellular base station [3]. Future D2D use cases include spectrum reuse, communication between UEs on opposite sides of the coverage boundary between two cells, and communication among UEs that are all outside any base station's coverage area and which therefore must be capable of self-organization [4, Fig. 1]. The last scenario is of particular interest to the public safety community [5].

An important D2D function is *discovery*, the process by which D2D UEs identify themselves to each other, using a specific set of time and frequency resources contained in a Physical Sidelink Discovery Channel (PSDCH). UEs that transmit announcements over the PSDCH randomly choose resources from a shared pool when they are not in the coverage of a base station, which can happen in public safety deployment scenarios, for example. Since multiple UEs can choose a given resource simultaneously, collisions between discovery messages may prevent them from being received by other D2D UEs. To address this problem, UEs use a transmission probability to throttle their use of the PSDCH; however, if this parameter is set too low, it will needlessly hinder UEs from quickly disseminating discovery information. We therefore need to find the optimal value for the transmission probability.

In this paper, we develop a closed-form expression for the optimal transmission probability that depends on the resource pool size, the number of subframes in the pool, and the number of UEs that use the pool. We discuss prior work in this area in Section II; we describe the SL discovery process in Section III, and then develop the mathematical model. In Section IV, we validate the model and examine its sensitivity to variations in the input parameters. We use Monte Carlo simulations to show that the optimal transmission probability also approximately minimizes the mean number of transmission periods for every UE to receive a discovery message from every other UE, which further demonstrates its usefulness. In Section V, we summarize our results and discuss future work.

II. Survey of Prior Work

Several recent studies have addressed issues related to discovery transmission; we describe these and their relation to our work in the following. Park *et al.* developed a scheme to reduce what they call the "participation delay," the time from a UE's entry to the network to its first beacon transmission [6]. They define temporary discovery resources, which are a small subset of the discovery resource pool and are reserved for the exclusive use of newcomers to the network. They show that this approach reduces the participation delay for new arrivals while having little impact on UEs already in the network. It is possible to use our model to optimize the size of the temporary pool, which we are undertaking as part of our ongoing work.

Kang and Kang used a stochastic geometry approach to determine how many devices can be discovered in a given number of PSDCH periods [7], but their approach is substantially different from ours. They modeled the population of devices using a non-homogeneous Poisson Point Process

Disclaimer: Certain commercial software packages are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the software packages identified are necessarily the best available for the purpose.

U.S. Government work not protected by U.S. copyright

Griffith, David; Lyons, Fiona.
"Optimizing the UE Transmission Probability for D2D Direct Discovery."
Paper presented at 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

(PPP) and used the Signal to Interference and Noise Ratio (SINR) to determine whether a signal could be received by a UE that was a given distance from the transmitting UE, using a simple path loss model that incorporates fading. While they considered the effect of half-duplex devices, they did not model the transmission probability.

Finally, the work by Bagheri *et al.* uses a model that obtains the probability of a collision in a given discovery resource [8]. However, their analysis fails to account for the half-duplex effect. They also do not consider the probabilistic transmission mechanism. Baghari *et al.*, like Kang and Kang, examines the effect of the SINR on the collision rate, assuming uniformly distributed UEs. We do not consider the possibility of a high SINR's allowing reception of a discovery message in this paper; instead, we use the more conservative assumption that collisions result in the loss of all colliding messages. We are including SINR and channel effects in an expanded model.

III. The Mathematical Model

A. The Sidelink Discovery Channel

Under Release 12, there are two D2D discovery modes for resource allocation [9, Clause 9.1.2]. Mode 1 discovery uses a base station to schedule and assign resources for transmitting discovery messages to UEs. With dedicated resources, UEs can operate without interference from other UEs that are associated with the same base station. In Mode 2, UEs autonomously choose resources from a pool; this reduces signaling but introduces the risk of collisions due to UEs' simultaneously randomly selecting the same resource. For out-of-coverage UEs, Mode 2 is the only possible option in the absence of a base station. The following analysis focuses on Mode 2 discovery.

The notation associated with the PSDCH and its attributes is listed in Table I. The PSDCH resource pool repeats periodically in the time domain; the period is given by the parameter P, which is set by the Information Element (IE) *discPeriod* [10, Clause 14.3.3]. The *discPeriod* IE is a parameter in the *SL-DiscResourcePool* IE, which is defined in [11, Clause 6.3.8].

Within each period, the PSDCH configuration variables *prb-Start*, *prb-End*, and *prb-Num*, which are respectively denoted by the variables $S1$, $S2$, and M, determine the range of subbands that the PSDCH occupies. The *SL-OffsetIndicator* IE gives the displacement of the pool from the first subframe in the period. The *SL-TF-ResourceConfig* IE [11, Clause 6.3.8] contains these IEs. A Physical Resource Block (PRB) with index m is part of the PSDCH if $S1 \leq m \leq S1 + M$ or $S2 - M \leq m \leq S2$. In the time domain, the set of subframes that compose the PSDCH is encoded in a bitmap defined by the IE *subframeBitmap*, where a bit that is set high indicates that the corresponding subframe is part of the PSDCH. The bitmap is repeated *numRepetition* times; this IE is located in the *SL-DiscResourcePool* IE [11, Clause 6.3.8]. In Fig. 1, we show the PSDCH structure and periodicity, and illustrate the role of the various PSDCH IEs.

A UE with a discovery message to send generates a uniformly distributed random value p1 $\in (0, 1]$. The UE sends its message if p1 is less than a threshold value *txProbability*, which is an IE within the *SL-DiscResourcePool* IE [12, Clause 5.15.1.1]. The *txProbability* IE can take one of the following four values: p25, p50, p75, and p100; these values denote thresholds of 0.25, 0.50, 0.75, and 1.00 respectively.

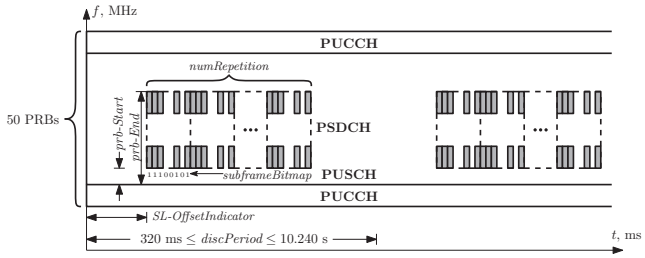


Fig. 1. The structure of the PSDCH resource pool.

UEs transmit discovery messages using pairs of PRBs in the PSDCH[1]. A transport block occupies two PRBs per slot in a single subframe [10, Clause 14.3.1]. The network operator can select the number of times each transport block is transmitted, $1 \leq N_{SLD}^{TX} \leq 4$. For Mode 2 SL discovery, the PRBs of the jth transmission in the ith PSDCH period are in subframe $l^{PSDCH}_{N_{SLD}^{TX} b_j^{(i)} + j - 1}$ and occupy the PRB indices $m^{PSDCH}_{2a_j^{(i)}}$ and $m^{PSDCH}_{2a_j^{(i)}+1}$, where

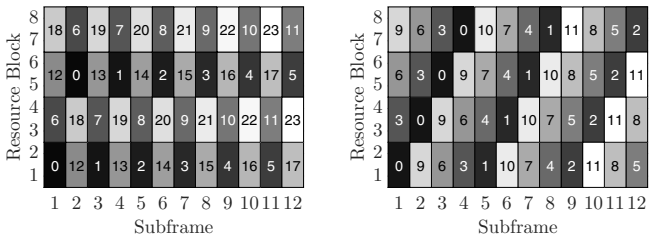
$$a_j^{(i)} = \left((j-1)\lfloor N_f / N_{SLD}^{TX} \rfloor + \lfloor n_{PSDCH} / N_t \rfloor\right) \bmod N_f \quad (1a)$$

$$b_j^{(i)} = n_{PSDCH} \bmod N_t. \quad (1b)$$

In Eq. (1), $N_f = \lfloor M_{RB}^{PSDCH\_RP} / 2 \rfloor$ is the number of PRB pairs in the frequency domain, where $M_{RB}^{PSDCH\_RP}$ is the number of PRBs in the PSDCH, $N_t = \lfloor L_{PSDCH} / N_{SLD}^{TX} \rfloor$ is the number of subframe sets in the time domain, where L_{PSDCH} is the number of subframes spanned by the PSDCH, and n_{PSDCH} is the resource index. The sender chooses n_{PSDCH} randomly from the set of $N_r = N_f N_t$ integers $\{0, 1, \ldots, N_r - 1\}$ [12, Clause 5.15.1.1]. If two UEs choose the same value of n_{PSDCH}, all N_{SLD}^{TX} of their transmissions will collide.

In Fig. 2, we show the placements of groups of PRB pairs for a hypothetical PSDCH consisting of $M_{RB}^{PSDCH\_RP} = 8$ PRBs in the frequency domain and $L_{PSDCH} = 12$ subframes in the time domain for $N_{SLD}^{TX} = 2$ and $N_{SLD}^{TX} = 4$ transmissions. Eq. (1) assigns resources to a set of transmissions such that the N_{SLD}^{TX} transmissions occur over a sequential series of subframe indices, and also over a sequential series of PRB pair indices, modulo N_f. Most importantly, the set of resource index values associated with a given subframe index i is identical to the set of resource index values in subframes $i + 1, i + 2, \ldots, i + N_{SLD}^{TX} - 1$. Thus, half-duplex UEs that choose values of n_{PSDCH} that produce identical values of $b_j^{(i)}$ will be unable to receive any of one another's transmissions for all values of j in a given PSDCH period, i. Because values of n_{PSDCH} that produce identical subframe

[1] A PRB spans twelve 15 kHz subcarriers and occupies a slot that is half of a 1 ms subframe.



(a) $N_{SLD}^{TX} = 2$ transmissions (b) $N_{SLD}^{TX} = 4$ transmissions

Fig. 2. Resources in PSDCH resource pool with $M_{RB}^{PSDCH\_RP} = 8$ PRBs and $L_{PSDCH} = 12$ subframes, from Eqs. (1a) and (1b).

TABLE I
LIST OF SYMBOLS

Symbol	Definition
$\mathcal{N}(A)$	Number of occurrences of event A
$\mathsf{Pr}\{A\}$	Probability that event A occurs
$\mathsf{E}\{Z\}$	Expected value of random variable Z
L_{PSDCH}	Number of subframes spanned by the PSDCH
$M_{RB}^{PSDCH\_RP}$	Number of PRBs occupied by the PSDCH
N_{SLD}^{TX}	Number of SL discovery transport block transmissions
l_i^{PSDCH}	ith subframe in the PSDCH, $0 \le i < L_{PSDCH}$
m_j^{PSDCH}	jth PRB slot in the PSDCH, $0 \le j < M_{RB}^{PSDCH\_RP}$
N_f	Number of PRB pairs in discovery pool
N_t	Number of subframe sets in discovery pool
N_r	Number of resources in discovery pool, $N_r = N_f N_t$
n_{PSDCH}	Discovery resource index
P	PSDCH period duration
θ	UE transmission probability
θ^*, θ_q^*	Optimal and quantized optimal values of θ
$\mathcal{G}$	The set of UEs in a D2D group of interest
N_u	Number of UEs in D2D group $\mathcal{G}$
X, Y	Randomly chosen UEs from D2D group $\mathcal{G}$
δ_X	Discovery message sent by UE X from D2D group $\mathcal{G}$
S_X	Set of subframes occupied by δ_X
$N_{Y\to X}$	Number of PSDCH periods for UE X to discover UE Y
$N_{\mathcal{G}\to\mathcal{G}}$	Number of PSDCH periods for discovery of all UEs in $\mathcal{G}$
$P_{Y\to X}(\theta)$	$\mathsf{Pr}\{$UE X receives UE Y's message in a given period$\}$
$\widehat{P}_{N_{Y\to X}}(\theta)$	Estimated value of $P_{Y\to X}(\theta)$
$\widehat{\sigma}^2_{\widehat{P}_{Y\to X}(\theta)}$	Sample variance of $\widehat{P}_{N_{Y\to X}}(\theta)$
$\Delta\mu_{Y\to X}$	Change in $\mathsf{E}\{N_{Y\to X}(\theta^*)\}$ due to quantizing θ
n	Index indicating the nth PSDCH period

assignments for a given transmission attempt produce identical subframe assignments for all other transmission attempts in the same PSDCH period. Thus, we can treat all N_{SLD}^{TX} subframes associated with a particular value of n_{PSDCH} as a single entity.

B. Model Description

We model the Discovery Resource Pool as a matrix with N_f rows and N_t columns that consists of $N_r = N_f N_t$ resources as shown in Fig. 3. Each row corresponds to a PRB pair, while each column corresponds to a subframe set. There are N_u UEs in group $\mathcal{G}$ that use the resource pool. Each UE transmits a discovery message during each period by choosing a resource in the pool with uniform probability. We assume that UEs choose resources independently of each other.

The $\times$'s in the boxes in Fig. 3 show the presence of discovery messages; the number of $\times$'s indicates the number of discovery messages. We assume that if more than one discovery message occupies a resource, then all of the co-located messages will be lost due to mutual interference. The $\otimes$ symbol indicates δ_X, the discovery message generated by UE X, and the column containing δ_X, which corresponds to S_X, is highlighted. Let S_X be the set of subframes occupied by UE X's discovery message. A half-duplex UE cannot transmit and receive at the same time, and thus misses any discovery messages that other UEs send in the subframes in which it is transmitting. Thus any discovery messages transmitted in S_X by other UEs in $\mathcal{G}$ will not be received by UE X.

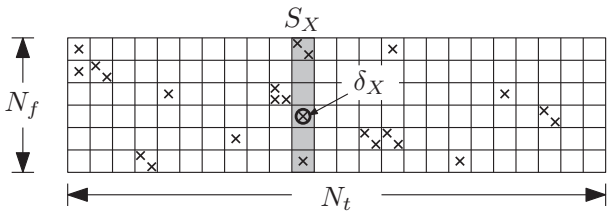


Fig. 3. A resource pool with $N_r = N_f N_t$ discrete resources organized into N_f frequency slots and N_t subframe sets, where S_X, the set of subframes used by UE X to transmit its discovery message, δ_X, is highlighted.

Let X and Y be two UEs chosen randomly from $\mathcal{G}$. Let $\{Y \to X\}$ be the event, "UE X successfully receives discovery message δ_Y from UE Y," and let $\{\delta_Y \in S_X\}$ be the event, "UE Y sends discovery message δ_Y in subframe S_X." A UE uses the same value of the parameter p1 to determine whether to transmit all N_{SLD}^{TX} of its transmission attempts in a PSDCH period. Thus, we can treat all of a period's transmissions by a UE as a single event and apply the resource grid that we introduced in Fig. 3.

Let $\theta = \mathsf{Pr}\{\text{p1} \le \textit{txProbability}\} = \mathsf{Pr}\{Y \text{ transmits}\}$. We define $P_{Y\to X}(\theta)$ in Table I; and it follows that $P_{Y\to X}(\theta) = \theta\,\mathsf{Pr}\{Y \to X \,|\, Y \text{ transmits}\}$. By Bayes' Theorem,

$$\begin{aligned}\mathsf{Pr}\{Y \to X \,|\, Y \text{ transmits}\} = \\ \mathsf{Pr}\{Y \to X \,|\, \delta_Y \in S_X\}\mathsf{Pr}\{\delta_Y \in S_X\} \\ + \mathsf{Pr}\{Y \to X \,|\, \delta_Y \notin S_X\}\mathsf{Pr}\{\delta_Y \notin S_X\}. \quad (2)\end{aligned}$$

In this case, the event $\{\delta_Y \in S_X\}$ occurs when UE X transmits (independently of all other UEs in the group, with probability θ) and UE Y's message δ_Y falls within the subframe set S_X. It follows that $\mathsf{Pr}\{\delta_Y \in S_X\} = \theta/N_t$, since UE X's decision to transmit and UE Y's choice of a subframe set are independent. As before, $\mathsf{Pr}\{Y \to X \,|\, \delta_Y \in S_X\} = 0$ since UE X and UE Y will not receive each others' message if they transmit in the same subframes. To determine $\mathsf{Pr}\{Y \to X \,|\, \delta_Y \notin S_X\}$, we condition on how many of the remaining $(N_u - 2)$ UEs transmit, which has a binomial distribution with probability mass function $f(k; N_u - 2, \theta) = \binom{N_u-2}{k}\theta^k(1-\theta)^{(N_u-2)-k}$. The probability a

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

UE that transmits does not use UE Y's resource is $(1-1/N_r)$. Thus Eq. (2) becomes

$$\begin{aligned}&\mathsf{Pr}\{Y \to X \,|\, Y \text{ transmits}\}\\ &\quad = 0 \cdot \tfrac{\theta}{N_t} + \left(1 - \tfrac{\theta}{N_t}\right) \sum_{k=0}^{N_u-2} \left(1 - \tfrac{1}{N_r}\right)^k f(k; N_u - 2, \theta).\end{aligned} \tag{3}$$

We recall that $P_{Y\to X}(\theta) = \theta\,\mathsf{Pr}\{Y \to X \,|\, Y \text{ transmits}\}$; we simplify Eq. (3) and apply the Binomial Theorem, and get

$$P_{Y\to X}(\theta) = \theta\left(1 - \frac{\theta}{N_t}\right)\left(1 - \frac{\theta}{N_r}\right)^{N_u-2}. \tag{4}$$

Assuming that the resource selection processes in different periods are independent, it follows that the number of periods for UE X to discover UE Y, $N_{Y\to X}(\theta)$, has a geometric distribution. Thus, the mean number of periods for UE X to discover UE Y is $\mathsf{E}\{N_{Y\to X}(\theta)\} = 1/P_{Y\to X}(\theta)$.

C. Optimizing the Transmission Probability

We can determine the value of θ that maximizes $P_{Y\to X}(\theta)$ by differentiating Eq. (4):

$$\begin{aligned}&\frac{\mathrm{d}P_{Y\to X}(\theta)}{\mathrm{d}\theta}\\ &\quad = \frac{(N_r-\theta)^{N_u-3}[N_r(N_t-2\theta)+\theta(N_t-N_tN_u+\theta N_u)]}{N_r^{N_u-2}N_t}.\end{aligned} \tag{5}$$

The derivative is zero when $\theta = N_r$, but this is not a valid solution. The polynomial in θ within the square brackets in the numerator of Eq. (5), $N_u\theta^2 - (2N_r - N_t + N_tN_u)\theta + N_rN_t$, is zero when

$$\theta = \frac{2N_r + N_t(N_u-1) \pm \sqrt{4N_r(N_r-N_t) + N_t^2(N_u-1)^2}}{2N_u}. \tag{6}$$

Because $N_t \le N_r$, the expression under the radical is always non-negative. Additionally, the positive branch is greater than or equal to unity[2]. Thus the negative branch of Eq. (6) gives θ^*, the optimal value of θ. We use the second derivative to verify that θ^* maximizes $P_{Y\to X}(\theta)$ as follows:

$$\ddot{P}_{Y\to X}(\theta^*) \overset{\text{def}}{=} \left.\frac{\mathrm{d}^2P_{Y\to X}(\theta)}{\mathrm{d}\theta^2}\right|_{\theta=\theta^*} = \frac{-A}{N_rN_t}\left(\frac{A+B}{2N_rN_u}\right)^{N_u-3}, \tag{7}$$

where $A = \sqrt{4N_r(N_r-N_t)+N_t^2(N_u-1)^2}$ and $B = (2N_r - N_t)(N_u - 1)$. Since $N_t \le N_r$, it follows that $A > 0$ and $B > 0$; thus, $\ddot{P}_{Y\to X}(\theta^*) < 0$ and $P_{Y\to X}(\theta^*)$ is a local maximum.

Depending on the relative values of the parameters N_r, N_t, and N_u, θ^* may fall outside the unit interval, i.e., $[0, 1]$. Expanding and simplifying the expression $\theta > 1$ using the negative branch of θ in Eq. (6) gives

$$N_u < \frac{N_r(N_t-2)+N_t}{N_t-1}. \tag{8}$$

[2]When $N_r = N_t = 1$, $\theta = 1$; and, the numerator of θ is greater than $2N_u$ when $1 < N_t \le N_r$.

Thus, θ^* should be set to unity when the number of UEs is sufficiently small for Eq. (8) to hold.

IV. Numerical Results

A. Model Validation

We validated this model by running Monte Carlo simulations to estimate $P_{Y\to X}(\theta)$. The simulation consisted of a set of runs; each run in turn was composed of a set of individual trials. Each trial simulated the UE group's resource selections over a series of PSDCH periods, using a given value of θ for all UEs in the group. In each period, the simulator generated a $N_f \times N_t$ matrix that represented the PSDCH resource pool, and randomly placed UE Discovery messages in the matrix.

The simulator determines whether a successful transmission from UE Y to UE X has occurred by examining the placement of UE X's transmission in relation to all other UEs. If UE X transmits in the same subframe set as UE Y, then UE X cannot receive UE Y's discovery message. If any of the other $(N_u - 2)$ UEs transmits in the resource block chosen by UE Y, then the transmissions collide and UE Y's message fails to reach UE X. The simulator also models the transmission probability θ, which is the same for all UEs, by having each UE generate a uniform random variate $0 < \mathrm{p1} \le 1$; if $\mathrm{p1} \le \theta$, then the UE chooses a resource for transmission. Thus UE Y's transmission fails if UE Y generates a variate that is greater than θ. Also note that if the variate that UE X generates is not less than θ, the transmission succeeds as long as no other UE's message occupies the resource chosen by UE Y.

The simulator repeated this process until the occurrence of a successful transmission, at which time it recorded $N_{Y\to X}(j)$, the number of PSDCH periods required to achieve the success in the jth trial, and started a new trial if there were trials remaining in the run. Using the set of trial results $\{N_{Y\to X}(j)\}_{j=1}^{N_{\text{trials}}}$, the estimate for $P_{Y\to X}(\theta)$ from the ith run is

$$\widehat{P}_{Y\to X,i}(\theta) = \frac{\mathcal{N}(N_{Y\to X}(\theta) = n)}{N_{\text{trials}}}, \tag{9}$$

where $\mathcal{N}(A)$ is the occurrence count of event A and N_{trials} is the number of trials per run. The estimated probability is

$$\widehat{P}_{Y\to X}(\theta) = \frac{\sum_{i=1}^{N_{\text{runs}}}\widehat{P}_{Y\to X,i}(\theta)}{N_{\text{runs}}} = \frac{\sum_{i=1}^{N_{\text{runs}}}\mathcal{N}(N_{Y\to X}(\theta)=n)}{N_{\text{runs}}N_{\text{trials}}} \tag{10}$$

and the estimator of the variance of $\widehat{P}_{Y\to X}(\theta)$ is

$$\widehat{\sigma}^2_{\widehat{P}_{Y\to X}(\theta)} = \frac{1}{N_{\text{runs}}-1}\sum_{i=1}^{N_{\text{runs}}}\left(\widehat{P}_{Y\to X,i}(\theta) - \widehat{P}_{Y\to X}(\theta)\right)^2. \tag{11}$$

The simulation used 10 runs, with 50 trials per run. The resource pool contained $N_r = 50$ resources, with $N_f = 10$ PRB pairs and $N_t = 5$ subframe sets. In Fig. 4, we plot both the theoretical values of $P_{Y\to X}(\theta)$ and the Monte Carlo estimates with 95 % confidence intervals for $N_u \in \{5, 10, \ldots, 50\}$ UEs[3]. We used the four enumerated values

[3]In some scenarios, the UE population may be larger, but this analysis can be readily extended to greater values of N_u and shows similar agreement between theoretical and simulation results.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

for *txProbability* for our θ values. The plots show good agreement between the theoretical and Monte Carlo results. Also, applying Eq. (8) indicates that we should set $\theta^* = 1$ when $N_u < 38.75$ UEs; the curves in the figure agree with this result. We examined other values of N_f and N_t and found similar agreement between the theoretical and Monte Carlo values; we do not show these plots due to space limitations.

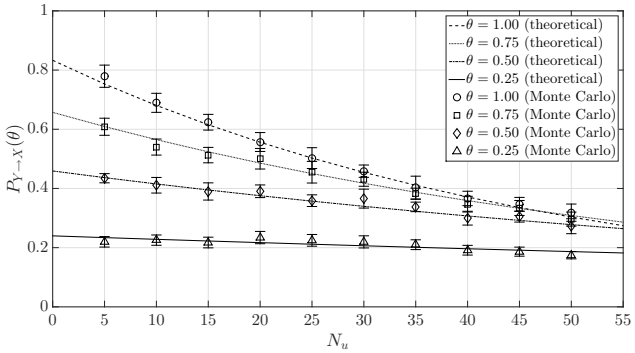


Fig. 4. Theoretical and simulated values of $P_{Y\to X}(\theta)$ plotted versus N_u, where $\theta \in \{0.25, 0.50, 0.75, 1.00\}$, for $N_f = 10$ PRB pairs and $N_t = 5$ subframe sets, with 95 % confidence intervals shown.

B. Sensitivity analysis

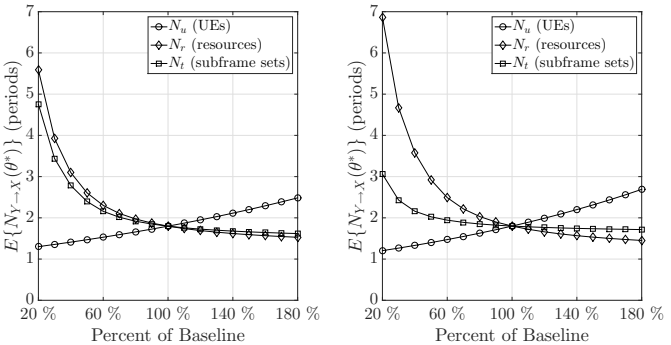
Next, we examine the effect of varying input parameters on the mean time to receive a discovery message. In Fig. 5, we show plots for two resource pool configurations, and examine the effect of varying N_r, N_t, and N_u. The greatest impact is due to reductions in N_r or N_t. $\mathsf{E}\{N_{Y\to X}(\theta^*)\}$ varies roughly linearly with respect to the percentage change in N_u.

We are especially interested in the impact of variations in N_u when θ^* has been chosen based on a particular value for the group size. In Fig. 6, we plot $\mathsf{E}\{N_{Y\to X}(\theta^*)\}$ versus N_u as a discrete sequence of points[4]. Next, for each value of N_u, we modify N_u by a fixed percentage, while keeping θ^* fixed, and recompute $\mathsf{E}\{N_{Y\to X}(\theta)\}$ using the new value of N_u. We plot the resulting sets of values for ± 10 % and ± 50 % variations in N_u in the figure; we show the envelopes traced by the modified values rather than discrete points for the sake of clarity.

The figure shows that $\mathsf{E}\{N_{Y\to X}(\theta^*)\}$'s sensitivity to variations in N_u increases as N_u itself increases, with a discontinuity in the slope of each envelope curve visible at $N_u = 38.75$ UEs, the threshold value given by Eq. (8). A ± 10 % deviation in N_u results in a variation of about half of a period when $N_u = 60$ UEs, and a variation of about one period when $N_u = 100$ UEs. When the variation is very large (± 50 %), increases in N_u have more effect than decreases.

Next we consider the impact of quantizing θ. In Section III we noted that θ can take only values that are multiples of $1/4$. Let $\theta_q^* = \max\big(1/4, \lceil 4(\theta^* - 1/8)\rceil/4\big)$ be the value of θ^* rounded to the closest allowed value of *txProbability*. In Fig. 7, we plot $\Delta\mu_{Y\to X} \stackrel{\text{def}}{=} \mathsf{E}\{N_{Y\to X}(\theta_q^*)\} - \mathsf{E}\{N_{Y\to X}(\theta^*)\}$ versus N_u, using the same resource pool dimensions as before. Since θ_q^* is suboptimal, $\mathsf{E}\{N_{Y\to X}(\theta_q^*)\} \geq \mathsf{E}\{N_{Y\to X}(\theta^*)\}$. Note that $\theta^* = 0.25$, 0.5, 0.75 when $N_u = 191$, 90, 56 UEs, respectively, and that $\theta^* = 1$ for $N_u < 39$ UEs; at these values of N_u, $\Delta\mu_{Y\to X} \approx 0$ periods. The discontinuities in Fig. 7 are products of the step discontinuities in the mapping that produces θ_q^* (e.g., $\theta_q^* = \frac{1}{2}$ for 70 UEs $\leq N_u \leq 123$ UEs but $\theta_q^* = \frac{1}{4}$ for $N_u \geq 124$ UEs). The figure shows that quantization introduces a penalty of at most half a period for most values of N_u; the penalty increases to a period only as N_u approaches 275 UEs. We examined other pool configurations and observed similar behavior; a rule of thumb appears to be that $\Delta\mu_{Y\to X} > 1$ period if $N_u \gtrsim 5N_r$.

[4]We chose $\mathsf{E}\{N_{Y\to X}(\theta^*)\}$ rather than its reciprocal, $P_{Y\to X}(\theta^*)$, because the effects of variations in N_u are easier to see in the plot.



(a) $N_r = 50$ $N_t = 5$, $N_u = 20$ (b) $N_r = 100$, $N_t = 10$, $N_u = 50$

Fig. 5. Spider plots for two example PSDCH configurations.

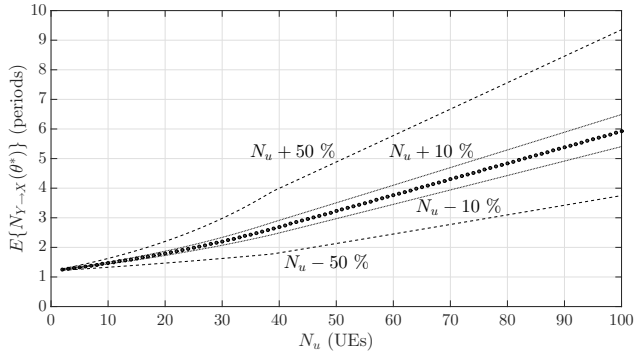


Fig. 6. Sensitivity plot of $\mathsf{E}\{N_{Y\to X}(\theta^*)\}$ versus N_u for $N_f = 10$ PRB pairs and $N_t = 5$ subframe sets, showing the effect of 10 % and 50 % errors in the value of N_u.

C. Impact of θ^* on group discovery time

Finally, we use Monte Carlo simulations to determine whether $\theta = \theta^*$ optimizes other performance metrics, particularly $\mathsf{E}\{N_{\mathcal{G}\to\mathcal{G}}(\theta)\}$, the mean number of periods for every UE in $\mathcal{G}$ to discover every other UE in $\mathcal{G}$. Each Monte Carlo trial used a $N_u \times N_u$ connectivity matrix to track the discovery status of each UE in $\mathcal{G}$. In each period, each UE chose a resource randomly, the ability of each UE to detect other UEs' messages was checked, and the connectivity matrix was updated. From these results we produced $\widehat{\mathsf{E}}\{N_{\mathcal{G}\to\mathcal{G}}(\theta)\}$.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

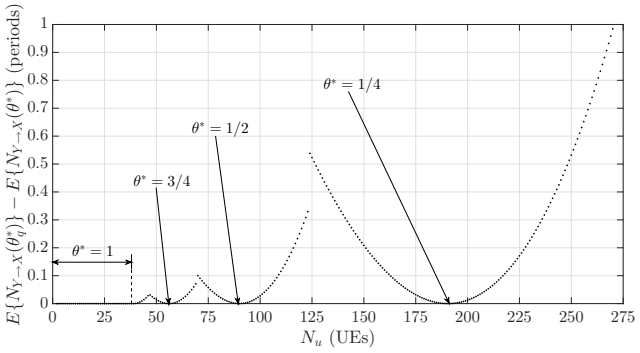
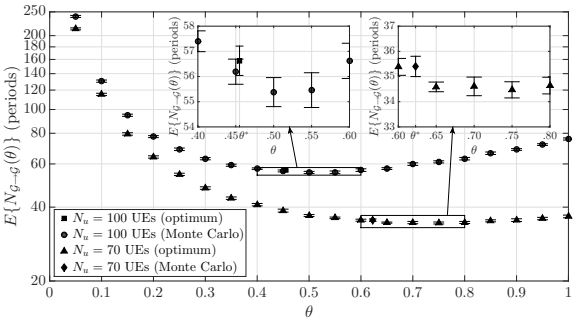
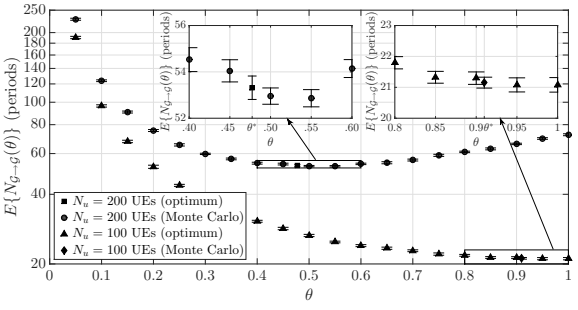


Fig. 7. Plot of $\mathsf{E}\{N_{Y\to X}(\theta_q^*)\} - \mathsf{E}\{N_{Y\to X}(\theta^*)\}$ versus N_u for $N_f = 10$ PRB pairs and $N_t = 5$ subframe sets, showing the effect of the quantization of θ.

Fig. 8, shows simulation results for two pool sizes, with $N_u = 100$ UEs and $N_u = 200$ UEs. We used 20 runs of 50 trials each to plot $\widehat{\mathsf{E}}\{N_{\mathcal{G}\to\mathcal{G}}(\theta^*)\}$ vs. θ, with 95 % confidence intervals. In each case, θ^* is close to the value of θ that minimizes $\mathsf{E}\{N_{\mathcal{G}\to\mathcal{G}}(\theta)\}$. Regarding the quantization of θ, θ_q^* tends to give the best possible discovery performance for the whole group, although there are exceptions such as the case $N_u = 70$ UEs as shown in Fig. 8a; in this case, the quantization gives $\theta_q^* = 0.5$, although $\theta_q^* = 0.75$ is the better choice. Our simulations have shown that θ^* tends to be less than the value of θ that optimizes group discovery performance, so rounding up to the next higher multiple of $1/4$ may give consistently near-optimal performance.



(a) $N_f = 10$ PRB pairs and $N_t = 5$ subframe sets



(b) $N_f = 10$ PRB pairs and $N_t = 10$ subframe sets

Fig. 8. $\widehat{\mathsf{E}}\{N_{\mathcal{G}\to\mathcal{G}}(\theta)\}$ versus θ, with 95 % confidence intervals shown.

V. Summary and Future Work

In this paper, we derived the optimal value for the UE transmission probability while accounting for the half-duplex nature of UE transmissions. We used this model to derive the maximum UE group size for which the optimal value of *txProbability* is unity. We validated this model and showed that there is a small impact to performance when quantizing θ^* to multiples of $1/4$, as allowed by the 3GPP standard. We demonstrated that θ^* appears to closely track the value of θ that minimizes the mean number of periods for all UEs in a group to discover each other, although this result needs further confirmation. Our next steps include accounting for channel effects in the model, and examining the effect of synchronization errors on PSDCH performance.

References

[1] D. Astely, E. Dahlman, G. Fodor, S. Parkvall, and J. Sachs, "LTE release 12 and beyond [accepted from open call]," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 154–160, July 2013.

[2] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, December 2009.

[3] X. Ma, R. Yin, G. Yu, and Z. Zhang, "A distributed relay selection method for relay assisted Device-to-Device communication system," in *IEEE 23rd Int. Symp. Personal Indoor and Mobile Radio Communications (PIMRC)*, September 2012, pp. 1020–1024.

[4] X. Lin, J. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3GPP device-to-device proximity services," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 40–48, April 2014.

[5] T. Doumi, M. Dolan, S. Tatesh, A. Casati, G. Tsirtsis, K. Anchan, and D. Flore, "LTE for public safety networks," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 106–112, February 2013.

[6] S. Park and S. Choi, "Expediting D2D discovery by using temporary discovery resource," in *2014 IEEE Global Communications Conf. (GLOBECOM)*, December 2014, pp. 4839–4844.

[7] H. Kang and C. Kang, "Performance analysis of device-to-device discovery with stochastic geometry in non-homogeneous environment," in *2014 Int. Conf. Information and Communication Technology Convergence (ICTC)*, October 2014, pp. 407–412.

[8] H. Bagheri, P. Sartori, V. Desai, B. Classon, M. Al-Shalash, and A. Soong, "Device-to-device proximity discovery for LTE systems," in *2015 IEEE Int. Conf. Communication Workshop (ICCW)*, June 2015, pp. 591–595.

[9] 3GPP, "Study on LTE Device to Device Proximity Services; Radio Aspects," 3rd Generation Partnership Project (3GPP), TR 36.843 V12.0.1, March 2014. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.843/36843-c01.zip

[10] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 3rd Generation Partnership Project (3GPP), TS 36.213 V12.7.0, September 2015. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.213/36213-c70.zip

[11] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.331 V12.7.0, September 2015. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.331/36331-c70.zip

[12] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.321 V12.7.0, September 2015. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.321/36321-c70.zip

Impact of timing on the Proximity Services (ProSe) synchronization function

Samantha Gamboa, Fernando J. Cintrón, David Griffith, Richard Rouil
National Institute of Standards and Technology, Gaithersburg, MD 20899-0001, USA
Email: {firstname.lastname}@nist.gov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

***Abstract*—Long Term Evolution Advanced (LTE-A) introduces a new feature called Proximity Services (ProSe) that enables device-to-device (D2D) communication between User Equipment (UE), including the capability to operate out-of-coverage. In order to establish a D2D communication link the UEs need to be synchronized. In out-of-coverage scenarios, the synchronization is performed in a distributed manner by the UEs. In this paper, we studied problems associated with the simultaneous execution of the synchronization procedure by LTE-A D2D-enabled UEs operating out-of-coverage. In particular, we focused on detection and convergence problems resulting from the half-duplex constraint and periodic scheduling. We showed that if two transmitter UEs are acting as synchronization references and they perform the procedure too close in time, convergence to a synchronized state is not possible. Moreover, the periodic triggering of the procedure will make the problematic condition persistent in time. We proposed an effective algorithm that prevent these problems, or resolve them in a reasonable time. We considered the protocol and requirements specified in the LTE-A standard, and we evaluated the performance of the proposed algorithm using system level simulations.**

I. Introduction

One new feature introduced by Long Term Evolution Advanced (LTE-A) is called Proximity Services (ProSe) [1]. ProSe enables device-to-device (D2D) communication between User Equipment (UE) via a direct link that has been given the term 'sidelink'. Additionally, the UEs can perform D2D communication in out-of-coverage conditions, i.e., without being attached to and controlled by an Evolved Node B (eNB). This is of vital importance for public safety uses [2], e.g., in emergencies or natural disasters causing network outage, or in mission critical interventions.

Most of the sidelink synchronization elements and procedures were derived from the LTE-A downlink design [3]. Each UE acting as a Synchronization Reference (SyncRef) transmits its synchronization information, which comprises several signals for frequency and time synchronization, and one information element containing system level information for further configuration.

Unlike an eNB, an out-of-coverage UE only acts as a SyncRef when transmitting on the sidelink [4]. Depending on the traffic pattern, a UE may transmit synchronization information only intermittently. Thus, a UE performing synchronization acquisition for a given period will only detect the SyncRef UEs that are actively transmitting during this period.

D2D-capable UEs use the same sidelink channel to transmit and receive. Given the half-duplex constraint, the UE has to switch between transmission and reception modes in order to avoid self-interference. This affects the synchronization process in the sense that a UE may not be able to perform the synchronization acquisition procedure and transmit data or its own synchronization signals at the same time. Thus, manufacturers need to design their synchronization procedures to be able to ensure a minimum level of synchronization performance while limiting the transmission drops.

A UE may perform downlink cell search for synchronization periodically, i.e., at regular intervals of time [5]. This is adequate, as the eNBs will persistently transmit the synchronization signals and information elements, and new cells will become available only due to UE mobility. In the D2D case, the UEs need to detect other UEs that are performing the same procedures in parallel. Moreover, UEs that are out-of-coverage rely only on preconfigured operational parameters, as there is no network coordination.

In this paper, we show that a periodic execution of the synchronization procedure in the out-of-coverage scenario can lead to SyncRef detection problems or a SyncRef ping-pong effect. The first issue occurs when two SyncRefs cannot detect each other because their synchronization procedures overlap in time, and the timely transmission and detection of each others synchronization signals is not possible. The second issue occurs when two SyncRefs are able to detect each other, but the synchronization executions are aligned in time such that each UE cannot perceive the change of condition of the other before making a synchronization decision. This will cause the SyncRefs to constantly synchronize to each other without converging to a single shared synchronization.

The rest of the paper is organized as follows. In Section II we provide a brief overview of the related literature. The sidelink synchronization procedure in out-of-coverage scenarios is described in Section III. In Section IV we define the system model and describe the identified problems. In Section V, we propose a strategy to cope with these problems, which is supported by the system level evaluation we present in Section VI. Section VII concludes the paper.

II. Related work

The LTE-A standard is specified by the 3rd Generation Partnership Project (3GPP). The sidelink synchronization signal design agreed by 3GPP is specified in [6], and Cannon et al. provide a comprehensive description in [3]. A UE should detect and successfully decode synchronization signals while

Cintron, Fernando; Gamboa Quintiliani, Samantha; Griffith, David; Rouil, Richard.
"Impact of timing on the Proximity Services (ProSe) synchronization function."
Paper presented at the 14th Annual IEEE Consumer Communications & Networking Conference,
Las Vegas, NV. January 8, 2017 - January 11, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

satisfying the performance requirements defined by 3GPP. However, the receiver design is left to UE implementation, and much of the sidelink synchronization literature focuses on the design of optimal receivers that satisfy these 3GPP requirements, e.g., [7] and references therein.

After a UE has detected one or more synchronization signals and identified one or more SyncRefs, it should synchronize to the most convenient one. Prior to the standardization agreements, several criteria were proposed in the literature for deciding which SyncRef is the most convenient. For example, Fodor et al. [8] proposed to use a weighting function that combines the device characteristics such as transmit power, battery level, network coverage and mobility as the metric used for the SyncRef decisions. Abedini et al. [9] proposed two approaches. If in-network synchronization information is detected, the metric is the number of hops away from the in-network SyncRef, so that the UE selects the SyncRef that is closer to the operated network. In the out-of-coverage case, the proposed metric is the age of the synchronization acquisition, and the UE selects the oldest synchronization information.

Finally, the synchronization protocol agreed by 3GPP defines three priorities based on the network coverage condition of the SyncRef. The highest priority goes to in-network SyncRefs, followed by out-of-coverage SyncRef at one hop of the network, and the lowest priority corresponds to fully out-of-coverage SyncRefs. The perceived signal strength is used as a tie-breaker between SyncRefs having the same priority [4].

How often a UE should look for, detect, and select an adequate SyncRef is left to UE implementation. However, 3GPP defines some related performance requirements the UE should meet [10]. To the best of our knowledge, our work is the first one to focus on problems related to the scheduling of the sidelink synchronization protocol, while considering the 3GPP specifications and performance requirements, which we explain in the next section.

III. Sidelink synchronization in out-of-coverage scenario

The out-of-coverage synchronization procedure comprises two different but related operations that the UE needs to perform. The first operation is related to the transmission of synchronization information. The UE needs to verify whether it has to become a SyncRef or not, and if so, which information it should broadcast and when. This operation is explained in Section III-A. The second operation is related to the acquisition of synchronization information. The UE needs to search for available SyncRefs and in case multiple SyncRefs are available, the UE needs to select the best one and synchronize to it. This operation is explained in Section III-B.

A. Sidelink synchronization information transmission

The decision of becoming a SyncRef depends on whether the UE has a selected SyncRef, i.e., the UE is synchronized to another transmitting UE and receiving synchronization information from it. If the UE does not have a selected SyncRef, it will become one itself. If the UE has a selected SyncRef, the decision depends on the selected SyncRef signal strength. The evaluation performed by the UE for taking this decision will be explained in Section III-B4.

A SyncRef uses the Sidelink Synchronization Signal (SLSS) for announcing its synchronization information. The SLSS is transmitted in one subframe in the time domain (i.e., a 1 ms time slot) and uses the central 6 resource blocks in the frequency domain. An SLSS is composed of four elements:

- Primary Sidelink Synchronization Signal (PSSS)
- Secondary Sidelink Synchronization Signal (SSSS)
- Demodulation Reference Signal (DMRS)
- Physical Sidelink Broadcast Channel (PSBCH)

The PSSS and SSSS together encode the SLSS identifier (SLSSID), which identifies the transmitted synchronization information. The PSBCH carries the *MasterInformationBlock-SL* (MIB-SL), which contains system level information needed for the configuration of the synchronizing UE [4]. The DMRSs are used as a reference for channel estimation, demodulation of the PSBCH and measurement of the Sidelink Reference Signal Received Power (S-RSRP) in the receiving UE.

The SLSS is sent with a periodicity of 40 ms. The exact time position is indicated by a preconfigured relative subframe offset. There are two preconfigured offsets (*syncOffsetIndicator1* and *syncOffsetIndicator2*) and the UE will choose one or the other depending on its synchronization condition.

B. Sidelink synchronization reference (re)selection

The SyncRef (re)selection process is done in four steps which we describe bellow. First, the UE performs a SyncRef search in order to find all available SyncRefs. Next, the UE performs S-RSRP measurements for each detected SyncRef in order to estimate the channel. Finally, the UE uses all the information gathered in the previous steps to decide to which SyncRef it will synchronize. Afterwards, the UE evaluates the selected SyncRef to determine if the UE itself has to become a SyncRef. We discuss the implications of the scheduling of this process at the end of this section.

1) SyncRef search: In this process the UE performs a full search for detecting the available SyncRefs. As the periodicity of the SLSS is 40 ms, the UE should search for at least this amount time in order to be able to detect at least one SLSS of each available SyncRef.

A SyncRef is considered detected by the UE if the UE has obtained the SyncRef SLSSID and has decoded the corresponding MIB-SL. The UE analyzes the detected signals as follows. First, a correlation with all the possible PSSS values is done. If a peak is detected, the associated sequence corresponds to the PSSS sequence, and the peak time position provides the subframe timing. Next, the SSSS sequence is identified by performing a correlation with all the possible SSSS values in the time position of the SSSS (relative to the PSSS). The PSSS and SSSS sequences are combined to obtain the SLSSID, which the UE uses to demodulate the PSBCH and obtain the MIB-SL.

Cintron, Fernando; Gamboa Quintiliani, Samantha; Griffith, David; Rouil, Richard.
"Impact of timing on the Proximity Services (ProSe) synchronization function."
Paper presented at the 14th Annual IEEE Consumer Communications & Networking Conference,
Las Vegas, NV. January 8, 2017 - January 11, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2) S-RSRP measurement: In order to estimate the channel between the UE and a given SyncRef, the UE measures the corresponding S-RSRP, which is defined as the linear average over the power contributions (in Watts) of the resource elements that carry DMRSs in the SLSS [11]. The UE applies two levels of filtering before using the S-RSRP measurements in any decision process:

a) Layer 1 (L1) filtering: At the physical layer, the UE takes several S-RSRP samples of the same SyncRef over a given period of time called the measurement period. The measurement period is defined as 400 ms, and the UE is allowed to measure up to 6 detected SyncRefs [10]. However, the number of samples taken by the UE is left to implementation, as long as the UE meets the associated accuracy requirements. These samples are averaged for a better estimation. At the end of the measurement period, the physical layer reports the averaged L1 S-RSRP to upper layers.

b) Layer 3 (L3) filtering: This is an optional process that uses an infinite impulse response filter to determine the S-RSRP quantity to be used by the decision process, i.e., the L3 S-RSRP. The filter is controlled by a preconfigured forgetting factor. If the factor value is high, the L3 S-RSRP value will be close to the most recent L1 S-RSRP value (i.e., the instantaneous channel state). If the factor value is low, the L3 S-RSRP value will consider the older L1 S-RSRP measurements (i.e., the history of the channel state).

3) Decision process: In the out-of-coverage scenario, all the UEs have the same priority, hence the selection of a SyncRef only depends on the set of SyncRefs S-RSRP values. The decision process uses the gathered information from the detected candidate SyncRefs: SLSSID, MIB-SL and (L3) S-RSRP. A SyncRef is considered valid if its S-RSRP is higher than a predefined minimum required threshold, which is specified in [10].

If the UE already has a valid selected SyncRef, the UE needs to check whether this SyncRef is still suitable, or if the SyncRef needs to be discarded and the UE needs to select a new SyncRef. The UE compares the selected SyncRef S-RSRP with the strongest S-RSRP SyncRef candidate. If the candidate exceeds the selected SyncRef S-RSRP by a preconfigured hysteresis value (*syncRefDiffHyst*), the selected SyncRef is discarded, and the UE goes through the selection process as if it did not have a SyncRef at the beginning of the procedure.

If the UE does not have a valid selected SyncRef, it will choose the valid candidate with the strongest S-RSRP.

4) Selected SyncRef evaluation: After selecting a SyncRef, the UE should determine whether it has to become or cease to be a SyncRef. This process should be performed within 0.8 s [10]. The UE should measure the S-RSRP of the selected SyncRef UE and take the decision. The UE will become a SyncRef if the S-RSRP of the selected SyncRef is below the threshold *syncTxThreshOoC* and it is transmitting sidelink communication. If any of these two conditions is not met, the UE will cease to be a SyncRef or will not become one.

5) Scheduling: Given the highly dynamic nature of D2D communication in out-of-coverage scenarios, the SyncRef (re)selection process has to be repeated often enough to minimize the data lost due to unsynchronized transmitter-receiver pairs. However, several of the processes needed for the SyncRef (re)selection (e.g., SyncRef search and L1 S-RSRP measurement) require the UE to be in receiving mode. This reduces the amount of time the UE will be able to perform transmissions, due to the half-duplex constraint. Considering this, the standard limits the time a UE can spend in receiving mode for performing SyncRef (re)selection as follows: in a period of 20 s, an out-of-coverage UE is allowed to drop a maximum of 2 % of its sidelink transmissions at the physical layer for the purpose of SyncRef (re)selection [10].

The details of the SyncRef search and S-RSRP measurement processes are left to implementation. However, we can see the practical implications of this constraint with the following example. Lets consider that a UE spends a total of 80 ms in receiving mode while performing the SyncRef (re)selection process (e.g., 40 ms for SyncRef search + 40 ms for L1 S-RSRP measurements). If we consider the theoretical worst case scenario in which the UE always has data to send, the UE may be able to use the whole 20 s for transmitting. In this case, the UE can only spend 400 ms in receiving mode for performing the SyncRef (re)selection process (i.e., 2 % of 20 s). Thus, the UE can perform at maximum 5 SyncRef (re)selection processes within the 20 s [1].

One simple way to schedule the SyncRef (re)selection process, is to distribute them in time with a fixed period. This technique is already used by commercial UEs for network-based synchronization, i.e., for downlink cell search and measurement [5]. In our example, the periodic SyncRef (re)selection process may occur every 4 s (5 times in 20 s), where the UE performs an initial SyncRef (re)selection process when it finds itself out-of-coverage (e.g., the UE turns ON and no eNB is available or the UE moves out-of-coverage), and repeats the process every 4 s.

However, using a fixed period in device-based synchronization may come with the cost of inflexibility. If the SyncRef (re)selection processes of two different UEs are aligned in time in a way that will not allow their mutual synchronization, this problem may continue during the entire communication session. In the next section, we characterize these problems, and in Section V, we propose an algorithm to address them. In Section VI, we evaluate both approaches using system level simulations.

IV. System model and problem definition

We consider an out-of-coverage scenario in which UEs arrive independently to the system according to a given arrival rate. A UE joining an out-of-coverage scenario can result from many different factors, e.g., the UE turned ON and no eNB was available, the UE moved to a zone without network coverage and lost network synchronization, the network suffered a blackout, or the in-network mode was simply

[1]This is the theoretical worst case scenario. If the UE is not transmitting 100 % of the time, more SyncRef (re)selection process can be scheduled.

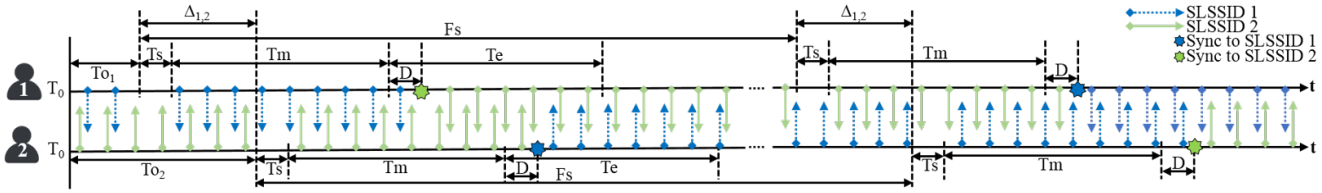


Fig. 1. Time alignment of the *periodic* SyncRef (re)selection process. SyncRef ping-pong effect example.

deactivated. When a UE arrives to the system, it performs an initial SyncRef (re)selection process. This process comprises a SyncRef search of duration T_s, a measurement period of duration T_m, and a decision process that we assume to be instantaneous. The effective change of timing and synchronization information occurs after a given delay, which is upper bounded by the parameter D. The UE evaluates the selected SyncRef during a period of duration T_e. Initially, we assume that the SyncRef (re)selection process will be repeated periodically after a given fixed time denoted as Fs.

In this scenario, we consider two transmitter UEs carrying different synchronization information. These UEs are mutually detectable at the time To, i.e., each UE is transmitting its own SLSSs and is receiving the SLSS of the other with a S-RSRP level above the minimum required after To. This can be the result of several cases. For example, the two UEs are already performing sidelink transmissions and they move into close proximity at time To; or the two UEs are already in proximity and they start their sidelink transmissions at To, or before To but after the last SyncRef (re)selection process of the other UE. Regardless of the case, the UEs do not detect each other before To but are mutually detectable after To.

The UEs will continue the sidelink transmissions for at least two SyncRef (re)selection periods. We denote as To_i the time relative to To in which UE i starts the next SyncRef (re)selection process. We define the parameter $\Delta_{i,j} = |To_i - To_j|$ to be the offset between the starting time of the SyncRef (re)selection processes of UE i and UE j. An example of the timeline of two UEs performing this periodical SyncRef (re)selection process is shown in Figure 1. We identified two possible synchronization problems in this scenario:

1) No detection: As both UEs are in proximity and advertising their synchronization information, they should be able to detect each other. However, due to the half-duplex constraint, if the UE is performing a SyncRef search, it cannot transmit SLSSs during that period. Thus, if the SyncRef search periods of the UEs are fully overlapped (i.e., $\Delta_{1,2} = 0$) the UEs will not be able to detect each other. Moreover, as Fs is fixed, this problem will persist. In the case that the SyncRef search periods of the UEs are partially overlapped (i.e., $0 < \Delta_{1,2} \leq T_s$), detection may be possible. This is the case if one of the UEs sends the SLSS before or after the SyncRef search period, and it is received in the non-overlapped part of the SyncRef search period of the other UE. Otherwise, the UEs will not be able to detect each other.

2) SyncRef ping-pong effect: If $\Delta_{1,2} > T_s$, the UEs will be able to detect each other. However, they may not be able to arrive to a common synchronization. For example, UE1 synchronizes to the information advertised by UE2 and UE2 synchronizes to the information advertised by UE1. In this case, as the period is fixed, this problem will be persistent as the UEs will change of synchronization information each time that the SyncRef (re)selection process is performed, without arriving to a synchronized state that allows them to communicate. We refer to this problem as the SyncRef ping-pong effect. It occurs when the synchronization events are aligned in a way that each UE cannot detect the change of synchronization information of the other UE, i.e., the SyncRef search of the UEs occurs before the decision process of the other UE finishes. In the described scenario, this happens when $0 < \Delta_{1,2} \leq T_s + T_m + D$, and Figure 1 depicts an example.

Table I shows a summary of the conditions and consequences explained above. We also include the condition in which the UEs will be able to converge to a synchronized state.

With this periodical algorithm, the value of $\Delta_{i,j}$ depends on the time in which the UEs arrive to the system, which is unknown by the other UEs. Thus, if a pair of UEs experiences any of the aforementioned problems, they will not have an exit condition provided directly by the synchronization protocol. An upper layer exit condition for the SyncRef ping-pong effect, can be that one of the UE stops to transmit, e.g., UE1, so that UE2 cannot detect UE1 anymore, and it keeps its

TABLE I
SUMMARY OF THE DIFFERENT POSSIBILITIES OF ALIGNMENT AND THE ASSOCIATED SYNCHRONIZATION CONDITIONS

Alignment	**Synchronization**	**Condition**	**Consequence**
$\Delta_{1,2} = 0$	No Convergence	No detection	The SyncRefs are not able to detect each other
$0 < \Delta_{1,2} \leq T_s + T_m + D$	Convergence Risk	No detection (Only if $0 < \Delta_{1,2} \leq T_s$)	The SyncRefs may not be able to detect each other
		SyncRef ping-pong effect	The SyncRefs may keep synchronizing to each other without converge to a synchronized state
$T_s + T_m + D < \Delta_{1,2} \leq F_s - (T_s + T_m + D)$	Convergence	None of the above	Convergence to a synchronized state

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Cintron, Fernando; Gamboa Quintiliani, Samantha; Griffith, David; Rouil, Richard.
"Impact of timing on the Proximity Services (ProSe) synchronization function."
Paper presented at the 14th Annual IEEE Consumer Communications & Networking Conference,
Las Vegas, NV. January 8, 2017 - January 11, 2017.

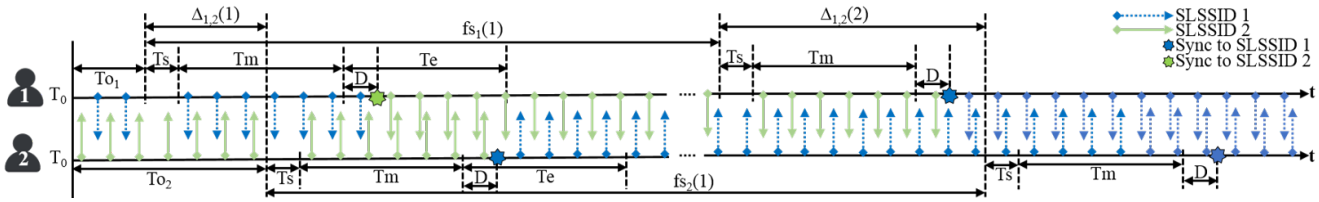


Fig. 2. Time alignment of the proposed *variable* SyncRef (re)selection process. SyncRef ping-pong effect resolution example.

own synchronization. Afterwards, UE1 will synchronize to the information of UE2 given that it is still transmitting. This will cause the convergence of both UEs to a synchronized state. However, this is not dependent on the synchronization protocol and it will not solve the detection problem.

V. Proposed Algorithm

In order to make the synchronization process resistant to the aforementioned problems, it should be decorrelated from the arrival of the UEs, and the persistent condition should be removed. To achieve these goals, we propose to trigger the SyncRef (re)selection process after a random backoff. Each time the process is performed, the UE schedules the next process to be triggered in a random time, i.e., the parameter Fs follows a random distribution.

We denote as $fs_i(k)$ the time between the k and $k+1$ SyncRef (re)selection process of UE i, i.e., $fs_i(k)$ is a realization of Fs and follows its distribution. Similarly, $\Delta_{i,j}$(m) denotes the time between the SyncRef (re)selection process of UE i and UE j at the m-th occurrence of the processes after To. Figure 2 shows an example of the timeline of two UEs using variable SyncRef (re)selection process triggering. The figure shows that $\Delta_{1,2}(1)$ and $\Delta_{1,2}(2)$ are different, which in this example removes the persistent ping-pong effect condition. This can be beneficial if any of the problems stated in Table I is encountered. For example, if $\Delta_{1,2}(1)$ satisfies any of the problematic conditions, it is less likely that $\Delta_{1,2}(2)$ falls under the same condition.

Moreover, the range of values for $fs_i(k)$ should be selected in order to maintain an adequate level of synchronization performance. The minimum value should be chosen to avoid that the UE performs the SyncRef (re)selection process too often, which may cause a percentage of transmission drops higher than the value established in the standard. The maximum value should be limited so that the UE performs the process often enough to react to synchronization changes. Finally, the length of the range should be large enough to ensure the variability needed for avoiding the synchronization problems.

VI. Evaluation

In previous work [12], we extended the LTE module of the ns-3 network simulator [13] to consider standard-compliant sidelink communications. The evaluations described in this section were performed using this implementation.

A. Scenario

The scenario is composed of three out-of-coverage UEs in proximity, i.e., they can detect each other and establish a communication session using the sidelink. Two of the UEs are interested in transmitting data to the third UE, i.e., there are two transmitters and one receiver. At the beginning of the evaluation, the UEs are not synchronized. The receiver will be able to receive the data from the two transmitters only after the three UEs are synchronized. Thus, a relevant metric for the scenario is the convergence time to a synchronized state, i.e., the time taken by the three UEs to acquire and use the same synchronization information.

The relevant parameters of the evaluation are summarized in Table II. Each UE i arrives at time T_i, and performs the initial SyncRef (re)selection process of the simulation at this time. The parameter b represents how scattered these arrivals can be within a SyncRef (re)selection process period. For simplicity of the evaluation, we assume the two transmitters start their transmissions at the same time $To = 10$ s, and the communication session lasts for the remaining simulation time. The simulation time was 70 s and the simulations were repeated 1000 times using different random seeds. The UEs wait until the end of its scheduler allocation period to apply the change of timing once a SyncRef is selected. Thus, D is in the worst case equal to the preconfigured allocation period, which is 40 ms.

TABLE II
Evaluation Parameters

Parameter	Value
Scenario	
T_i (ms)	$Unif(0,b) \vert b \leq fs_{\text{min}}$
Synchronization protocol	
syncTxThreshOoC (dBm)	-60
syncRefDiffHyst (dB)	0
syncOffsetIndicator[1,2]	7, 3
Layer 3 filtering	Deactivated
SyncRef (re)selection process parameters	
T_s (ms)	40
T_m (ms)	400
T_e (ms)	400
D (ms)	40
Maximum time in receiver mode	
For SyncRef search (ms)	40
For Measurement (ms)	36
For Evaluation (ms)	4
Time in Rx mode Total (ms)	80
SyncRef (re)selection process triggering	
Periodic: F_s (ms)	fs_{min}
Variable: F_s (ms)	$Unif(fs_{\text{min}}, fs_{\text{max}})$
Evaluations performed	
fs_{min} (ms)	Evaluation A: 4000 Evaluation B: 2000 Evaluation C: 1000
fs_{max} (ms)	$(1+\alpha) * fs_{\text{min}} \mid \alpha \in [0,1]$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Cintron, Fernando; Gamboa Quintiliani, Samantha; Griffith, David; Rouil, Richard.
"Impact of timing on the Proximity Services (ProSe) synchronization function."
Paper presented at the 14th Annual IEEE Consumer Communications & Networking Conference,
Las Vegas, NV. January 8, 2017 - January 11, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

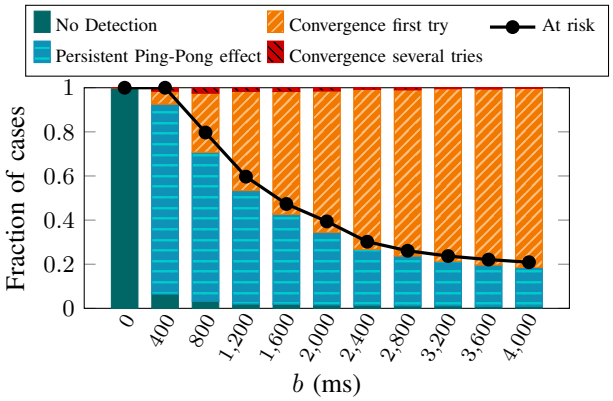


Fig. 3. Distribution of the synchronization conditions depending on the UE arrival distribution ($T_i \sim Unif(0, b)$) when using the *periodic* algorithm in the Evaluation A.

B. Algorithm configuration

The parameter fs_{min} is the minimum value Fs could take to satisfy the transmission drop rate constraint depending on the UE transmission conditions (Section III-B5). We performed three different evaluations (A, B and C) based on the values of fs_{min} (See Table II). The value fs_{min} = 4000 ms (Evaluation A) was obtained considering the theoretical worst case scenario (i.e., UE transmits 100 % of the time), and considering that the UE needs to spend 80 ms in reception mode for performing the SyncRef (re)selection process (Table II). The parameter fs_{min} can be adapted depending on the UE traffic conditions, the scheduling policies of both control and traffic channels, and the actual number of SyncRefs the UE detected.

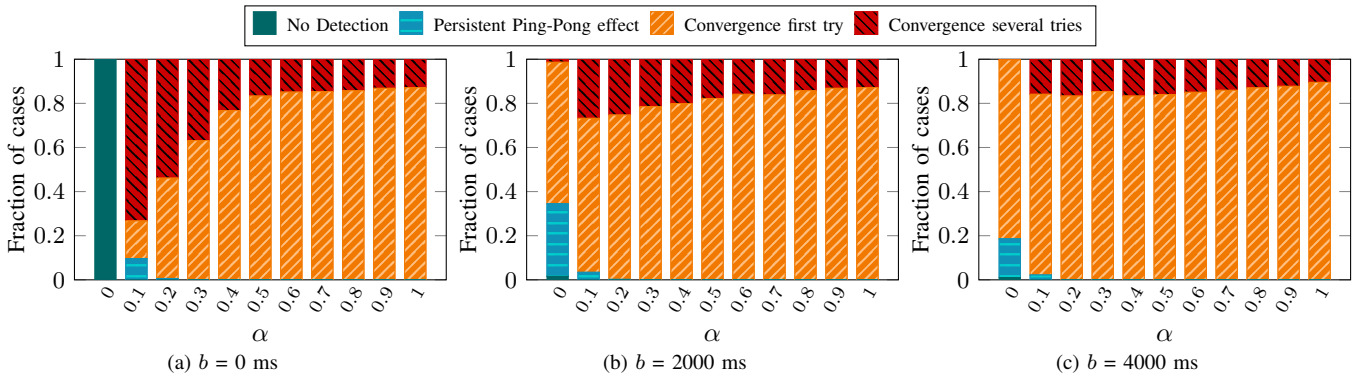
Each UE performs the SyncRef (re)selection process each fs_{min} ms when using the *periodic* algorithm. With the proposed *variable* algorithm, each UE schedules the next process in a time randomly chosen between fs_{min} and fs_{max}. We vary the value of fs_{max} using the parameter α (Table II) in order to explore the performance of the proposed algorithm.

C. Results

Figure 3 shows the distribution of the synchronization conditions for Evaluation A when the UEs are using the periodic algorithm. When b = 0 ms, all the UEs arrive at the same time. This implies that all UEs perform the initial process at the same time ($T_1 = T_2 = T_3 = 0$), and therefore they perform every process at the same time. This prevents the UEs from detecting each other, and synchronization convergence is not possible. The percentage of synchronization convergence increases with b, i.e., the more scattered the arrivals, the fewer synchronization problems are observed. However, in the best case (b = 4000 ms), we observe that 18 % of the cases still encountered a synchronization problem. The same trend is observed in Evaluation B and C.

The fraction of cases at risk in Figure 3 corresponds to the cases with a (re)selection process time alignment favorable for synchronization problems, i.e., the value of $\Delta_{1,2}$ was in one of the problematic alignment intervals in Table I. However, not all the cases that were at risk actually experienced a synchronization problem, or had one persistently. For example, with b = 400 ms, 8 % of the cases at risk converged to a synchronized state, while with b = 2000 ms 13 % of the cases at risk converged. Synchronization was achieved in these cases when the alignment of the (re)selection processes was favorable for convergence, either in the first synchronization attempt or after several ones. Initial convergence was due to partially overlapped SyncRef search periods, as explained in Section IV. Convergence after several tries was observed when the SyncRefs experienced the ping-pong effect, but the condition resolved. For example, after the SyncRefs synchronize to each other, each one changes its time reference, which also modifies the timeslot in which the SLSSs are sent. After one or several of these changes, the alignment can be favorable for convergence. However, Figure 3 shows that convergence after several tries happens rarely, as it depends on the alignment of multiple factors such as the choice of the offsets, the exact moment of transmission of SLSSs, how unsynchronized the UEs were upon arrival to the system, and the timing change history.

Figure 4 shows the results for Evaluation A when considering the proposed variable algorithm. The value $\alpha = 0$ corresponds to the periodic algorithm. The data show that with a value of $\alpha = 0.1$, the variable algorithm can considerably reduce synchronization problems, e.g., from 100 % to 10 % when b = 0 ms, from 35 % to 4 % with a b = 2000 ms, and from 18 % to 2 % when b = 4000 ms. Moreover, synchronization



(a) b = 0 ms (b) b = 2000 ms (c) b = 4000 ms

Fig. 4. Distribution of the synchronization conditions depending on α when using the *variable* algorithm in the Evaluation A. Three different values of b are considered and please note that the value $\alpha = 0$ corresponds to the *periodic* algorithm.

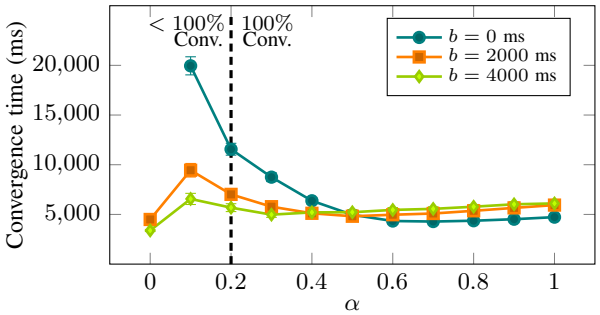


Fig. 5. Convergence time when using the *variable* algorithm in the Evaluation A. Only the cases where convergence was achieved are considered.

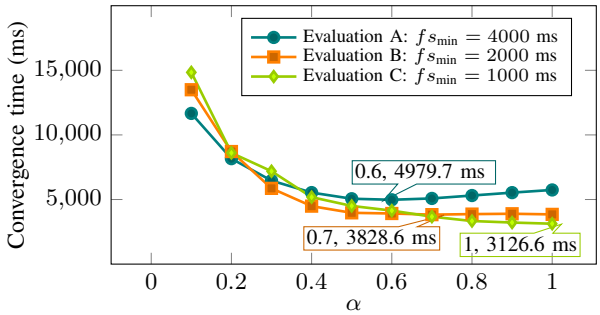


Fig. 6. Average convergence time when using the *variable* algorithm. The optimal values are highlighted with data labels for each evaluation.

problems can be completely avoided by choosing $\alpha \geq 0.2$ in most of the cases, which highlights the need for and benefits of using a variable algorithm.

Figure 5 shows the convergence time as a function of α. This figure shows the average and the 95 % confidence interval considering only the cases where convergence was achieved. Note that when b = 0 ms and $\alpha = 0$ there is no convergence, as shown in Figure 4a. Note also that the minimum value of convergence time for b = 2000 ms and b = 4000 ms corresponds to $\alpha = 0$, however, a non-negligible percentage of the evaluated cases did not converge in these scenarios: 35 % and 18 %, respectively (Figure 4). In the cases where 100 % of convergence was achieved, we observe that the convergence time has an upward concave shape as a function of α, reaching its minimum value in different α values depending on the parameter b, e.g., $\alpha = 0.7$ for b = 0 ms, $\alpha = 0.5$ for b = 2000 ms and $\alpha = 0.4$ for b = 4000 ms (Figure 5). However, the UEs do not have prior knowledge of the value of b, and the parameter α should be selected in order to provide the overall best performance considering all values of b.

Figure 6 shows the average convergence time, which is calculated considering all simulated values of b. This figure shows the results for the different evaluations (i.e., different values of $fs_{\min}$). Note that the smaller the value of $fs_{\min}$, the more often in average the UE will perform the SyncRef (re)selection process. However, the average convergence time is not necessarily smaller when $fs_{\min}$ decreases, as shown in Figure 6 for Evaluation C and $\alpha < 0.7$. The reason is that with small values of $fs_{\min}$ and α the algorithm is not able to create enough variability in the triggering of the SyncRef (re)selection process, which causes the need for the UEs to perform the process several times to finally arrive to a synchronized state. However, with $\alpha \geq 0.7$, Evaluation C exhibits the lower average convergence time, which highlights the importance of choosing an adequate α for the variable algorithm.

VII. Conclusion and Future Work

In this paper, we studied the system level implications of the device-based synchronization protocol for LTE-A D2D-enabled UEs operating out-of-coverage. We showed that performing the synchronization reference (re)selection process periodically, i.e., at fixed intervals of time, can lead to convergence problems in scenarios when two synchronization references are simultaneously active. We proposed an effective technique that allows to avoid these problems, or to resolve them in a reasonable time if they occur. The solution is based on a random backoff to trigger the synchronization reference (re)selection process. We showed using system level simulations the trade-offs that should be considered for achieving a given level of performance and satisfy the 3GPP standard requirements.

In future work, we will consider the design and evaluation of algorithms that can dynamically adapt the triggering of the process depending on the UE transmission conditions, e.g., the traffic status, the scheduler configuration, etc.

References

[1] 3GPP, "Technical Specification Group Services and System Aspects; Proximity-based services (ProSe); Stage 2 v.12.7.0," 3rd Generation Partnership Project (3GPP), TS 23.303, 2015.

[2] S. Y. Lien, C. C. Chien, F. M. Tseng, and T. C. Ho, "3GPP device-to-device communications for beyond 4G cellular networks," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 29–35, March 2016.

[3] M. Cannon, "On the Design of D2D Synchronization in 3GPP," in *IEEE ICC 2015 - Workshop on Device-to-Device Communication for Cellular and Wireless Networks*, 2015.

[4] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification v.12.8.0," 3rd Generation Partnership Project (3GPP), TS 36.331, 2016.

[5] J. Salo and J. Reunanen, "Interlayer Mobility Optimization," in *LTE Small Cell Optimization: 3GPP Evolution to Release 13*. John Wiley & Sons, Ltd., 2016.

[6] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation v.12.8.0," 3rd Generation Partnership Project (3GPP), TS 36.211, 2016.

[7] A. K. R. Chavva and K. Sripada, "Low-complexity LTE-D2D Synchronization Algorithms," in *IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2016, pp. 156–163.

[8] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi, "Device-to-device communications for national security and public safety," *IEEE Access*, vol. 2, pp. 1510–1520, 2014.

[9] N. Abedini, S. Tavildar, J. Li, and T. Richardson, "Distributed Synchronization for Device-to-Device Communications in an LTE Network," vol. 15, no. 2, pp. 1547–1561, 2016.

[10] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management v.12.12.0," 3rd Generation Partnership Project (3GPP), TS 36.133, 2014.

[11] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer - Measurements v.12.2.0," 3rd Generation Partnership Project (3GPP), TS 36.214, 2015.

[12] R. Rouil, F. Cintron, A. Ben Mosbah, and S. Gamboa, "An LTE Device-to-Device module for ns-3," in *2015 Workshop on ns-3 (WNS3)*, 2016.

[13] NS-3 Consortium, "ns-3 Network Simulator," Available at: https://www.nsnam.org/.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Cintron, Fernando; Gamboa Quintiliani, Samantha; Griffith, David; Rouil, Richard.
"Impact of timing on the Proximity Services (ProSe) synchronization function."
Paper presented at the 14th Annual IEEE Consumer Communications & Networking Conference,
Las Vegas, NV. January 8, 2017 - January 11, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Wideband Spectrum Reconstruction with Multicoset Sub-Nyquist Sampling and Collision Classification

Raied Caromi, Jae-Kark Choi, Wen-Bin Yang and Michael Souryal
Communications Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
Email: {raied.caromi, jae-kark.choi, wen-bin.yang, michael.souryal}@nist.gov

***Abstract*—This paper proposes an improved method for reconstructing wideband sparse spectrum. We utilize a multicoset setup based on time delay. The simple multicoset setup is more suitable for practical implementation in comparison to more sophisticated sub-Nyquist systems. We first introduce the general reconstruction model that solves for a fixed number of variables. We employ a simple machine learning technique to classify the aliased sub-Nyquist bins into two categories. The classification method reduces the reconstruction time by decreasing the number of combinations and variables needed for resolving the signals. The saving in solution time is significant at low occupancy levels. Furthermore, the approach is robust against higher noise levels, because although the classification accuracy decreases as SNR decreases, the reduction in the accuracy of the classifier does not adversely affect the overall detection. We define detection performance metrics and provide simulation results to demonstrate the effectiveness of our approach.**

I. Introduction

A key challenge for spectrum sharing systems is the need for spectral occupancy information over a wideband. In addition, spectrum sharing techniques require sensing results from a large geographic area. In general, it is difficult to collect instantaneous spectrum occupancy information unless multiple parallel narrow-band, or wideband sensors are employed. However, it may not be very cost-effective to use wideband sensors, or many conventional sensors in parallel. Moreover, high sampling rates result in high energy consumption. Various sub-sampling techniques are proposed for wideband spectrum sensing [1]. Usually, these techniques require the spectrum to be sparse. The sparsity condition is often satisfied when the spectrum is viewed from a wideband perspective. The sparsity assumption is even more accurate in rural areas.

Numerous approaches have been developed for wideband spectrum sensing and reconstruction. The majority of the compressive sensing approaches for spectrum sharing are less practical, or require special hardware designs. Some of these techniques require random sampling, and analog processing [2], [3]. Other techniques require co-prime sampling rate combinations that can not be achieved with conventional ADCs [4]. A practical implementation of wideband compressed sensing is studied in [5]. A wideband sensing approach called BigBand is proposed in [6]. Specifically, BigBand is based on multicoset sampling and it is implemented using conventional analog to digital converters (ADCs). While BigBand approach is a good candidate for practical implementation due to its simplicity and fast reconstruction time, some issues were still not addressed. For instance, BigBand relies on the phase rotation property to detect a change in magnitude and decides if there is a collision between Nyquist bins that fall into the same sub-Nyquist bin. However, if the multiband signal is corrupted by noise, there will be a change in magnitude even when no collision occurs. This is especially true at low signal to noise ratio (SNR) scenarios. Furthermore, BigBand suggests to set all the frequencies that are associated with a specific sub-Nyquist bin to be occupied when a solution is not realized. Therefore, higher false positive rates are expected. In general, false positives in spectrum sensing techniques are not as harmful as false negatives. However, if the false positives are spread over a wide band and separated by the sub-Nyquist rate, then large chunks of the spectrum will be rendered as not usable for many spectrum sharing techniques.

This paper focuses on the development of a low complexity spectrum sensing and reconstruction technique that is suitable for practical implementation with conventional ADCs. We adopt a multicoset uniform-sampling setup with time delays similar to the one in [6]. We develop the general least square model to reconstruct the wideband signals under the condition of sparsity. In order to reduce the total reconstruction time, we incorporate a simple classification model to classify the sub-Nyquist bin collision order. The collision classification will enable us to solve for lower number of combinations. Hence, the total solution time is reduced. The final reconstruction algorithm combines the least square resolver and the collision detector. Since most spectrum sharing techniques require only the occupancy measurements, we focus on detection performance metrics to evaluate our system. However, the accuracy of signal reconstruction is another possible metric for evaluation. For instance, the estimation error of the magnitude and the phase of the reconstructed signal can be evaluated but is beyond the scope of this paper and will be considered in future work. The main contributions of this paper are: a) the solution for the general signal reconstruction case is developed, and b) a classifier is utilized to reduce solution time while keeping minimum effect on the reconstruction accuracy along wide range of SNR values.

The remainder of this paper is organized as follows. Section II describes the model and presents the sub-sampling approach. Section III provides the steps for reconstructing the sub-

U.S. Government work not protected by U.S. copyright

Caromi, Raied; Choi, Jae-Kark; Souryal, Michael; Yang, Wen-Bin.
"Wideband Spectrum Reconstruction with Multicoset Sub-Nyquist Sampling and Collision Classification."
Paper presented at 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

sampled signal, presents the collision classifier, and defines the performance metrics. Section IV demonstrates the simulation results. Finally, Section V concludes the paper.

II. System Model and Problem Statement

Let $x(t)$ be a wideband signal of interest that is bandlimited to $[-\frac{F_N}{2}, \frac{F_N}{2}]$, where F_N is the required Nyquist sampling rate that guarantees a full recovery of the signal. Practically, $x(t)$ is the sum of different signals and noise over a multiband channel. Let $x[n] = x(t = nT_N), n = 0, \ldots, N-1$ be the Nyquist sampled version of $x(t)$, where $x \in \mathbb{C}^N$ and the sampling time $T_N = 1/F_N$. The frequency domain representation of $x[n]$ can be calculated by computing the discrete Fourier transform (DFT) of $x[n] \overset{DFT}{\longleftrightarrow} X = [X_0\ X_1\ X_2\ \ldots\ X_{N-1}]^\intercal$. Given that X is sparse in the frequency domain, we wish to recover the signal $x(t)$ using a sampling rate that is less than the Nyquist sampling rate F_N.

We utilize a multicoset system with uniform samplers and time delays. Fig. 1 shows the multicoset system setup. The input signal $x(t)$ is delayed over M branches, each with a time delay of τ_m. To simplify the analysis, we restrict the time delay to multiples of T_N, i.e., $\tau_m = \eta_m T_N$, $\eta_m \in \mathbb{Z}_0^+$. The delayed signal is uniformly sub-sampled with a sampling rate $F_S = \frac{1}{T_S} < F_N$, and $y[\ell] = y(t = \ell T_S), \ell = 0, \ldots, L-1$ is the sub-sampled version of the input signal at branch m. The DFT of $y^m[\ell]$ is denoted by Y^m. The sub-samplers in this setup should not employ anti-aliasing filters; the aliased signals are later resolved in signal reconstruction. The down sampling ratio is defined as $D = \frac{F_N}{F_S} = \frac{N}{L}$, where L is the length of $y^m[\ell]$ and is equal to the number of sub-Nyquist bins. To justify such a system in practice, MF_S must be less than F_N (otherwise, we could divide the band into M sub-bands, each of which is Nyquist-sampled); thus, $M < D$. Furthermore, assume F_N is integer multiples of F_S. Since $x(t - \tau_m)$ is sampled with a sampling rate $F_s < F_N$, the sampled signal is aliased and the corresponding Nyquist bins fall into specific sub-Nyquist bins. Consider M sampling branches as shown in Fig. 1. For each branch m, the sub-Nyquist bin ℓ can be written in terms of Nyquist bins as

$$Y_\ell^m = \sum_{i=0}^{D-1} X_{(\ell+i*L)} e^{-j2\pi(\ell+i*L)\frac{\eta_m}{N}}, \quad (1)$$
$$\ell \in \{0, 1, 2, \ldots, L-1\}, m \in \{0, 1, 2, \ldots, M-1\},$$

where $(\ell+i*L)$ is the index of the bin frequency $f_{(\ell+i*L)}$. The phase shift term $e^{-j2\pi(\ell+i*L)\frac{\eta_m}{N}}$ is the result of the time shift and can also be represented in terms of frequency and time delay as $e^{-j2\pi f_{(\ell+i*L)}\tau_m}$. Fig. 2 demonstrates the mapping of the frequency bins from the Nyquist dimension to the sub-Nyquist dimension. Clearly, one sub-Nyquist bin Y_ℓ is equal to the summation of X_n values in row ℓ.

In practice, the sampled signal will be corrupted by noise. Hence,

$$x(t) = x_s(t) + x_w(t), \quad (2)$$

where $x_s(t)$ is the signal part, including any channel distortion, and $x_w(t)$ is additive white Gaussian noise (AWGN). In addition, let κ be the set of indices of occupied Nyquist bins, or more specifically, the Nyquist bins that contain signal energy. The number of occupied Nyquist bins is given by $|\kappa| \leq N$, and the occupancy level is defined as $(\frac{|\kappa|}{N} * 100)\%$. If the Nyquist bandwidth is sparse enough, some of the sub-Nyquist bins Y_ℓ^m may contain only noise energy. Thus, if we know this information for a specific sub-Nyquist bin, we can declare D corresponding Nyquist bins as unoccupied. To simplify the problem further, we rewrite the sub-Nyquist bin as a combination of signal and noise components.

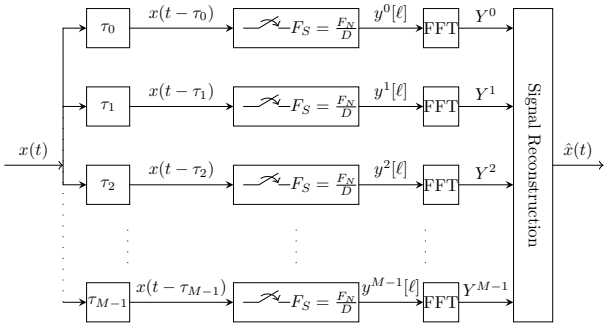


Fig. 1. Delay-based multicoset sampling.

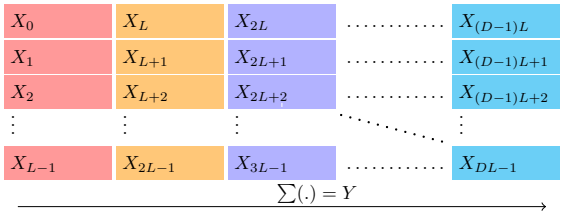


Fig. 2. Frequency bin mapping from Nyquist to sub-Nyquist dimension.

$$Y_\ell^m = \sum_{i=0}^{D-1} \left(X_{s(\ell+i*L)} + X_{w(\ell+i*L)}\right) e^{-j2\pi(\ell+i*L)\frac{\eta_m}{N}}, \quad (3)$$
$$X_s = 0 | (\ell + i*L) \notin \kappa,$$
$$\ell \in \{0, 1, 2, \ldots, L-1\}, m \in \{0, 1, 2, \ldots, M-1\}.$$

Equation (3) implies that for a given ℓ if $X_{s(\ell+i*L)} = 0, \forall i \in \{0, 1, 2, \ldots, D-1\}$, then D corresponding bins contain only noise. The first step to reconstruct the original signal is to identify the sub-Nyquist bins that contain at least one non zero signal bin. This can be achieved by a simple threshold-OR rule

$$\mathcal{L}(\ell) = \begin{cases} 1 & \text{, if } (|Y_\ell^0|^2 > \gamma) \vee \cdots \vee (|Y_\ell^{M-1}|^2 > \gamma), \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where $\mathcal{L}$ is the set of threshold decisions, and γ is the threshold. For each sub-Nyquist bin that contains signal energy, we try to resolve that specific bin to its Nyquist equivalent. Nevertheless, the reconstruction task is not trivial since we don't know how many Nyquist bins are active in that specific bin. The assumption in [6] exploits the phase rotation property between different branches to decide whether there is a collision between Nyquist bins in that specific sub-Nyquist bin. This is accomplished by detecting a change in magnitude

Caromi, Raied; Choi, Jae-Kark; Souryal, Michael; Yang, Wen-Bin.
"Wideband Spectrum Reconstruction with Multicoset Sub-Nyquist Sampling and Collision Classification."
Paper presented at 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC. December 4, 2016 - December 8, 2016.

among the branches. However, once the signal is corrupted with noise, there will always be a change in magnitude among the branches. Furthermore, a higher noise level leads to a higher change in magnitude.

Since we are not interested in estimating Nyquist bins that contain only noise, we define frequency collision as the case when more than one Nyquist bin that contains signal energy falls into the same sub-Nyquist bin. In addition, we define the collision order of a sub-Nyquist bin as the number of Nyquist bins that contain signal energy and map to that specific sub-Nyquist bin. Let $c_\ell \in \mathbb{Z}_0^+$ be the sub-Nyquist bin collision order, i.e., $c_\ell = 0$ means the sub-Nyquist bin contains only noise, $c_\ell = 1$ means the signal energy in a sub-Nyquist bin originates from one specific Nyquist bin, and $c_\ell \geq 2$ means that two or more Nyquist bins that have signal energy fall into one sub-Nyquist bin. In general, highly sparse spectrum leads to more sub-Nyquist bins of order zero and one. On the other hand, resolving one sub-Nyquist bin results in estimating D Nyquist bins. Consider the case when only two components in a specific sub-Nyquist bin contain signal energy and the rest are only noise, e.g., $m = \{0, 1, 2\}$, $\ell = 0$, and $\{2L, 6L\} \in \kappa$.

$$\begin{aligned} Y_0^0 =& X_{(2L)} e^{-j2\pi(2L)\frac{\eta_0}{N}} + X_{(6L)} e^{-j2\pi(6L)\frac{\eta_0}{N}} \\ Y_0^1 =& X_{(2L)} e^{-j2\pi(2L)\frac{\eta_1}{N}} + X_{(6L)} e^{-j2\pi(6L)\frac{\eta_1}{N}} \\ Y_0^2 =& X_{(2L)} e^{-j2\pi(2L)\frac{\eta_2}{N}} + X_{(6L)} e^{-j2\pi(6L)\frac{\eta_2}{N}} \end{aligned} \tag{5}$$

We are interested in estimating the two main signal components $X_{(2L)}$ and $X_{(6L)}$. In general, there is no exact solution to (5). The system of equations in (5) is overdetermined and consists of a linear combination of the variables $X_{(2L)}$ and $X_{(6L)}$. Typically, the number of variables that exist and contain signal energy is not known. Without loss of generality, we will restrict the solution to an overdetermined system that is one dimension higher than the number of variables. Therefore, the maximum number of possible equations is equal to the number of branches M.

The overdetermined system of (5) can be solved using the least square method. However, in practice we don't know which and how many variables exist. To address this issue, we can solve for all possible combinations of the frequencies for a given number of equations and variables. Furthermore, while it is difficult to detect a frequency collision in the presence of noise, the noise would also make it difficult to resolve the case of no collision without solving for the system of equations. Consider the case of one variable when $\eta_0 = 0$, e.g.,

$$\begin{aligned} Y_0^0 =& X_{(2L)} \\ Y_0^1 =& X_{(2L)} e^{-j2\pi(2L)\frac{\eta_1}{N}}. \end{aligned} \tag{6}$$

Ideally, if we know that only one Nyquist bin that contains signal energy falls into Y_0^m, which is similar to the case of no collision in [6], then we can easily calculate the frequency given that $\angle Y_0^1 - \angle Y_0^0 = -2\pi f_{(\ell+i*L)}\tau_1$. However, due to the effect of noise, the value of the calculated frequency will not be accurate. In addition, it will be difficult to accurately decide where the detected bin falls when the frequency resolution is increased. Hence, we will also use the least square method to solve for the case of no collision.

III. Signal Reconstruction

Let the number of variables for sub-Nyquist bin ℓ be $v_\ell \leq c_\ell$, and the number of equations $u_\ell \leq M$. For M branches, (1) can be written in a matrix form as

$$\boldsymbol{Y_\ell} = \boldsymbol{A_\ell} \boldsymbol{X_\ell}, \tag{7}$$

where $\boldsymbol{Y_\ell} \in \mathbb{C}^{u_\ell \times 1}$, $\boldsymbol{A_\ell} \in \mathbb{C}^{u_\ell \times v_\ell}$, $\boldsymbol{X_\ell} \in \mathbb{C}^{v_\ell \times 1}$. For each sub-Nyquist bin, $\boldsymbol{X_\ell}$ consists of v_ℓ variables out of D possible variables. Hence, we want to find the best $\boldsymbol{X_\ell}$ fit among all possible combinations. In the following subsection, we define a generalized form for this problem.

A. Least Square Solution

Define $\mathcal{X}_\ell$ as the set of all combinations of $\boldsymbol{X_\ell} \in \mathbb{C}^{v_\ell \times 1}$ from D possible variables. Specifically, $\mathcal{X}_\ell := \{\boldsymbol{X_\ell^1}, \boldsymbol{X_\ell^2}, \ldots, \boldsymbol{X_\ell^{|\mathcal{X}_\ell|}}\}$. Similarly, define the set of all corresponding phase shift matrices $\mathcal{A}_\ell := \{\boldsymbol{A_\ell^1}, \boldsymbol{A_\ell^2}, \ldots, \boldsymbol{A_\ell^{|\mathcal{X}_\ell|}}\}$, where $|\mathcal{X}_\ell| = \binom{D}{v_\ell}$. At each ℓ, $\boldsymbol{Y_\ell} \in \mathbb{C}^{u_\ell \times 1}$ is the same for any combination. For each combination of variables $\boldsymbol{X_\ell^j}$, define the residual as

$$\begin{aligned} \boldsymbol{r_j} =& \boldsymbol{Y_\ell} - \boldsymbol{A_\ell^j} \boldsymbol{X_\ell^j}, \\ & \boldsymbol{X_\ell^j} \in \mathcal{X}_\ell, \boldsymbol{A_\ell^j} \in \mathcal{A}_\ell. \end{aligned} \tag{8}$$

We want to minimize $\boldsymbol{r_j}$ for a suitable choice of $\boldsymbol{X_\ell^j}$. The general minimization problem becomes

$$\underset{\boldsymbol{X_\ell^j} \in \mathcal{X}_\ell}{\operatorname{argmin}} \left\| \boldsymbol{r_j} \right\|_2. \tag{9}$$

In general, $\boldsymbol{r_j}$ cannot be made zero. In addition, (9) can be reduced to $|\mathcal{X}_\ell|$ individual problems. Furthermore, for each value of j, the problem is the well defined least square which is optimal for the linear model. Therefore, the solution is to find $\boldsymbol{X_\ell^j} \in \mathbb{C}^{v_\ell \times 1}$ such that $\left\| \boldsymbol{Y_\ell} - \boldsymbol{A_\ell^j} \boldsymbol{X_\ell^j} \right\|_2$ is minimized, $\forall j \in \{1, 2, \ldots, |\mathcal{X}_\ell|\}$. The pseudoinverse solution for each combination is given by

$$\hat{\boldsymbol{X}}_\ell^{\boldsymbol{j}} = (\boldsymbol{A_\ell^j}^{\mathsf{H}} \boldsymbol{A_\ell^j})^{-1} \boldsymbol{A_\ell^j}^{\mathsf{H}} \boldsymbol{Y_\ell},$$

which can be obtained by solving the normal equations or by using the QR decomposition method [7]. The final step is to select $\hat{\boldsymbol{X}}_\ell^{\boldsymbol{j}}$ with minimum $\left\| \boldsymbol{r_j} \right\|_2$. In principle, solving for more variables when there are enough branches produces a better estimation of the Nyquist bins. This is true even when there are fewer Nyquist bins that originate from a signal in comparison to the number of variables. However, this may increase the complexity of the problem. For instance, the order of change in the number of combinations between v_ℓ and $v_\ell + 1$ for the same compression ratio is equal to $(D - v_\ell)/(v_\ell + 1)$, e.g., if $D = 10$, and $v_\ell = 2$, then the number of combinations for v_ℓ is equal to 45, and for $v_\ell + 1$ is equal to 120. Therefore, we can reduce the solution time by solving for fewer variables, which results in a trade-off between accuracy and time.

B. Collision Classifier

Although we are interested in estimating bin collision order, collision detection should be sufficient enough to reduce the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

computational complexity under the assumption of sparse spectrum. We consider a similar concept of phase rotation as in [6], except we no longer assume the change in magnitude to be zero in the case of no collision. We propose a collision detection scheme based on machine learning. Specifically, the algorithm aims to predict the collision occurrence by utilizing a trained model. The changes in magnitude among branches are attributes supplied to the algorithm. As such, α_ℓ^m is defined as the change in magnitude between branch 0 and m. Namely,

$$\alpha_\ell^m = |Y_\ell^m| - |Y_\ell^0|, \forall m \in \{1, \ldots, M-1\}, \tag{10}$$

and $\boldsymbol{\alpha_\ell} = [\alpha_\ell^1, \alpha_\ell^2, \ldots, \alpha_\ell^{M-1}]^\intercal$.

We adopt the Naïve Bayes classifier (NBC) as a collision detector because of its high speed[1]. Let $p(c_\ell|\boldsymbol{\alpha_\ell})$ be the probability of the observation $\boldsymbol{\alpha_\ell}$ being in class c_ℓ. From Bayes rule we have

$$p(c_\ell|\boldsymbol{\alpha_\ell}) = \frac{p(c_\ell)p(\boldsymbol{\alpha_\ell}|c_\ell)}{p(\boldsymbol{\alpha_\ell})}, \tag{11}$$

in which $p(c_\ell)$ is the prior probability of class c_ℓ. It is difficult to compute $p(\boldsymbol{\alpha_\ell}|c_\ell)$ unless conditional independence of the attributes given the class is assumed. Although this assumption is generally not satisfied, NBC still results in a classifier that often performs well [8]. Hence, the class conditional probability is given by

$$p(\boldsymbol{\alpha_\ell}|c_\ell) = \prod_{m=1}^{M-1} p(\alpha_\ell^m|c_\ell) \tag{12}$$

The normalization value in (11) is the same regardless of the class and can be ignored. The correct class can be computed by using a maximum a posteriori (MAP) estimator.

$$\underset{\text{MAP}}{c_\ell} = \underset{c_\ell}{\operatorname{argmax}}\ p(c_\ell)p(\boldsymbol{\alpha_\ell}|c_\ell) \tag{13}$$

C. Reconstruction Algorithm

Although the proposed classifier can identify different collision orders with acceptable accuracies, special attention should be given to its effect on the overall detection performance. As such, carefully chosen cost functions should be incorporated with the classifier to give a much higher cost for falsely predicting lower order collisions. In addition, since our objective is to reconstruct a highly sparse spectrum, most of the saved time comes from identifying no collision cases, i.e., $c_\ell = 1$. Therefore, we aim to predict two states of collision order, i.e., $c_\ell = 1$, and $c_\ell \geq 2$. The signal reconstruction approach is shown in the following algorithm. Before the algorithm enters the working mode, the classifier needs to be trained. The training process can be achieved by generating different signals in an environment that is representative of the overall band. The Nyquist and sub-Nyquist versions of these signals are collected and fed to the classifier for training.

In practice, the classifier can be trained by capturing signals from the real environment of interest. The sub-Nyquist samplers can still be used to capture Nyquist signals within their limits and hop over multiple chunks of the wide band. The training process is only required for one time, and subsequently the trained classifier can be used to detect collisions. For each sub-Nyquist bin that is not free, we test for collision detection. If no collision is predicted, we solve for D cases of one variable. Otherwise, if collision is detected, we solve for $\binom{D}{v_\ell}$ cases of v_ℓ variables.

Algorithm Signal reconstruction algorithm

INITIALIZE Trained collision detector, v_ℓ, M, N, L
  Sample signal from M branches
  $D \leftarrow \frac{N}{L}$
  Evaluate Y^m for M branches
  for $\ell = 0$ to $L-1$
    if $\mathcal{L}(\ell)$
      Collision detector: c_ℓ?
      if $c_\ell = 1$
        $v_\ell \leftarrow 1$, $u_\ell \leftarrow 2$
        Solve $|\mathcal{X}_\ell| = D$ combinations of one variable
      else if $c_\ell \geq 2$
        $v_\ell \leftarrow M-1$, $u_\ell \leftarrow v_\ell + 1$
        Solve $|\mathcal{X}_\ell| = \binom{D}{v_\ell}$ combinations of v_ℓ variables
      end if
      $X \leftarrow \begin{cases} \hat{\boldsymbol{X}}_{\boldsymbol{\ell}}^{\boldsymbol{j}} & \text{, if minimum } \|\boldsymbol{r_j}\|_2 \text{,} \\ 0, & \text{otherwise.} \end{cases}$
    else
      $X(\ell + i * L) \leftarrow 0, \forall i = \{0, 1, \ldots, D-1\}$
    end if
  end for

D. Performance Metrics

System performance depends on many factors such as sparsity, SNR, threshold and the accuracy of the classifier. In order to evaluate the performance of the reconstruction algorithm, we generate multiple sinusoids randomly distributed over the entire band. Each sinusoid has a random amplitude and phase within some range. We combine these sinusoids and propagate them over a frequency-selective block-channel fading with AWGN. We set one sampling block to NT_N in which the channel does not change. This configuration will enable us to control the occupancy level over a wideband channel while measuring different performance metrics. Recall that κ is the set of indices of occupied Nyquist bins; consequently, it represents the set of active sinusoids. Therefore, the number of active sinusoids is equal $|\kappa|$. However, some sinusoids may be undetectable due to the combined effect of the randomization of their amplitudes and the frequency selectivity of the channel. Therefore, we evaluate probability of detection P_D, and probability of false alarm P_{FA} relative to a typical Nyquist energy detection. Denote by CPD, the number of correct positive decisions. The detection ratio DR is defined as

$$\text{DR} = \frac{\text{CPD}}{\text{NCPD}}, \tag{14}$$

where NCPD is the number of Nyquist CPD, i.e., the number of correct detection decisions in the Nyquist bandwidth with-

[1]It should be noted that other classifiers such as support vector machine and K-nearest neighbors may outperform NBC in terms of accuracy.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

out sub-sampling. We also define the false alarm ratio FR as

$$\mathrm{FR} = \frac{\mathrm{FPD}}{N - \mathrm{NCPD}}, \tag{15}$$

where FPD is the number of false positive decisions.

We average DR and FR over many Monte-Carlo iterations to estimate P_D and P_{FA}. The upper limit for P_{FA} for this approach is equal to $\frac{v_\ell}{D}$. This stems from the fact that detection decisions are made for v_ℓ variables out of D Nyquist bins for each sub-Nyquist bin, regardless if they are correct decisions, we declare the remaining $D - v_\ell$ as free. The reason for choosing this approach is twofold. First, there is no robust method to decide whether the least square estimation was the correct one aside from the minimum of residuals approach proposed in Section III-A. Second, even if we impose a threshold-based decision on the residuals to dismiss some decisions, declaring D Nyquist bins that are spread over the whole bandwidth as occupied will render large chunks of the spectrum as not useable.

The overall algorithm can be viewed as a two stage system. The first stage is the collision detector, and the second stage is the signal resolver. Occupancy detection is evaluated at the second stage, but its performance is also affected by the first stage. Our goal is to reduce the computational complexity of the signal resolver by correctly detecting collisions. The reduction in the computation steps is crucial for practical implementation since it reduces execution time. At the same time, we want to ensure minimum effect on the accuracy of detection. Obviously, when the collision detector falsely predicts a collision, the overall detection error is mostly not affected. By contrast, when the collision detector predicts no collision while in fact there is a collision, the overall detection error is highly affected. Hence, there is a trade-off between accuracy and computational complexity.

IV. Performance Evaluation

This section presents system performance results. A multi-sinusoid test signal with a specific occupancy level is generated at each sampling block. Sinusoid frequencies are randomly chosen from the set of all possible frequencies in the band. In addition, tone amplitudes and phases are chosen from $\mathcal{U}(0.6, 1)$, and $\mathcal{U}(-\pi, \pi)$ respectively. The overall wideband signal is propagated over a frequency-selective Rayleigh fading channel that is constant during one sampling block. The channel tap delays and average path gains are set to $(0,\ 1,\ 2,\ 4)$ ns, and $(0,\ -0.5,\ -1.5,\ -2)$ dB, respectively. The Rayleigh fading process is normalized such that the average value of the path gains' total power is equal to one. The multi-sinusoid signal combined with the effect of the channel represents x_s in (2). In addition, $x_w \sim \mathcal{CN}(0, \sigma_w^2)$ in (2) is added to x_s, where σ_w^2 is the variance of x_w, to generate the total multiband signal $x(t)$. We define the SNR to be equal to $1/\sigma_w^2$. All simulations are performed under the assumption of 1 GHz total bandwidth, while each branch is capable of sampling at a 100 MHz sampling rate. The time shift factors were chosen as follows, $\eta_0 = 0$, $\eta_1 = 1$, $\eta_2 = 2$, and $\eta_3 = 3$. Furthermore,

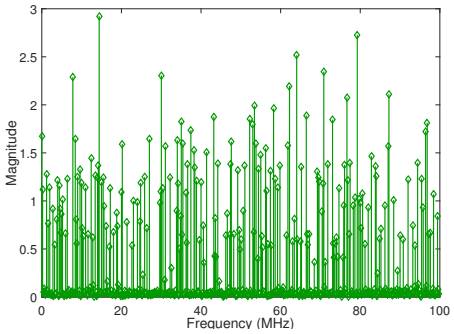


Fig. 3. Sub-sampled signal, 5% occupancy, SNR=0 dB, $\gamma = 0.1$.

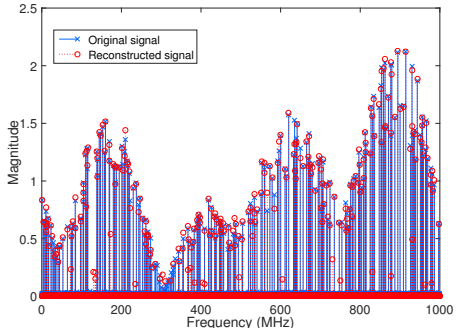


Fig. 4. Reconstructed signal with DR $= 0.978$, and FR $= 0.069$.

$N = 5000$, $L = 500$, and $D = 10$. An example of a sub-sampled signal is demonstrated in Fig. 3, and its reconstructed version along with the original signal are shown in Fig. 4. Due to the fact that this is a relatively high-SNR, low-occupancy signal, a three variable resolver is able to reconstruct it with high detection ratio.

A. Detection Performance

Figs. 5 and 6 show the performance results of Monte-Carlo simulations with a fixed two-variable resolver for different occupancy cases. Two of the simulated cases are carefully selected to show the effect of having a resolver with higher or equal number of variables in comparison to the collision order. Namely, in the first case listed in the legend, the tones are generated to produce no collision and each sub-Nyquist bin contains only one tone. On the contrary, all the sub-Nyquist bins contain two tones in the third case. Although we solve for two variables in both cases, P_D is better in the first case in comparison to the third case. The higher P_D in the first case comes at the cost of higher P_{FA}. This is due to the fact that we use two-variable resolver to resolve collisions of order one. Specifically, each solution may yield a false alarm since the two variable resolver gives a solution with two components and one of them is not occupied. This example demonstrates that the overall error probability bounds for this scheme are sophisticated and generally depend on delay set combinations, threshold values, and the order of the resolver relative to the average number of collision types. The tones in the second and the fourth cases are randomly placed in the entire bandwidth with 10% and 20% occupancies, respectively. The fourth case perform worse than the second case because it has significantly

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

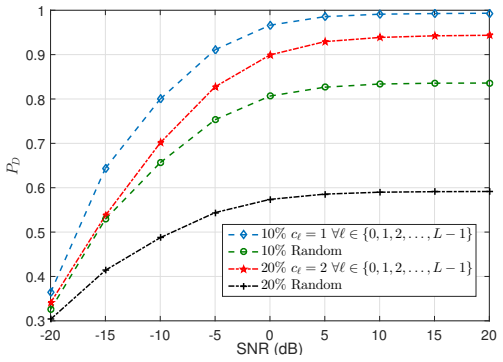


Fig. 5. P_D vs SNR for different occupancy cases, $\gamma = 0.05$.

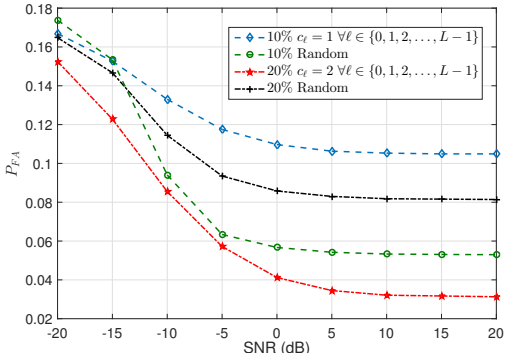


Fig. 6. P_{FA} vs SNR for different occupancy cases, $\gamma = 0.05$.

larger number of collisions of third and higher order, which produce more of both false alarms and mis-detections.

Fig. 7 and Fig. 8 show the detection performance of two- and three-variable resolvers with respect to occupancy level. Higher occupancy levels produce higher-order collisions. As a result, performance degrades as occupancy increases due to a higher number of unresolved signals. As expected, the three variables resolver can maintain higher detection ratios for more values of high occupancy in comparison to the two variables resolver. The improvement comes at the cost of requiring more branches and spending more time to solve for more combinations. While probability of detection is of highest priority for spectrum sensing, reconstruction latency can be considered one of the biggest obstacles when implementing sub-Nyquist sampling schemes. Therefore, any improvement in computation time is useful for practical implementation.

B. Collision Detector

The first step for implementing the reconstruction algorithm in Section III-C is to train the collision classifier described in Section III-B. We use three branches to generate training data based on the parameters provided in Section IV for multiple levels of sparsity and SNRs. The training data, meant to represent a practical scenario, is then used to train the classifier. Fig. 9 shows the confusion matrix of a trained classifier, where TN is the number of true negatives, FP is the number of false positives, FN is the number of false negatives, and TP is the number of true positives. The false positive rate $\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$, and the false negative

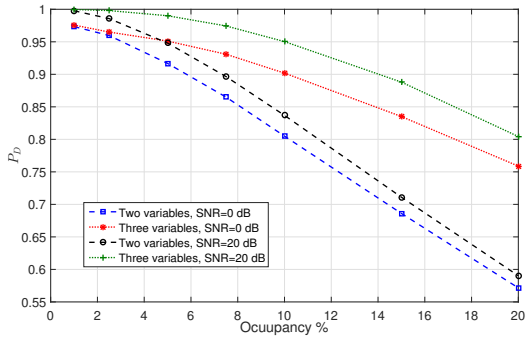


Fig. 7. P_D vs occupancy for two and three variables, $\gamma = 0.05$.

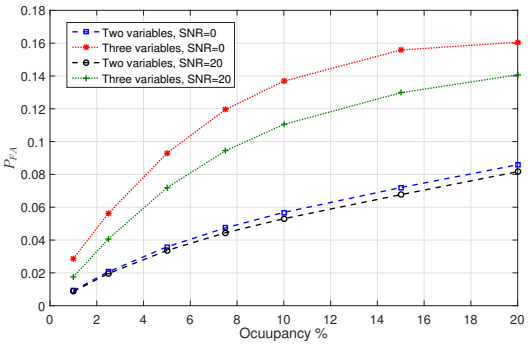


Fig. 8. P_{FA} vs occupancy for two and three variables, $\gamma = 0.05$.

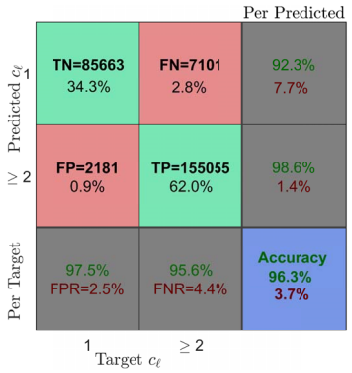


Fig. 9. Confusion matrix of collision detector, SNR=20 dB.

rate $\mathrm{FNR} = \mathrm{FN}/(\mathrm{FN} + \mathrm{TP})$. The accuracy of the classifier is defined as $(\mathrm{TN} + \mathrm{TP})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})$.

In general, FPR value of the classifier does not affect the overall signal detection because the resolver will use more variables to resolve the collision. As a result, it is of our interest to make the classifier more biased towards false positives. The bias can be imposed by either adding a cost function to the classifier with a higher cost given for FN decisions, or by training the classifier at a higher SNR than average. The latter method will set the classifier to produce more FP and less FN at low SNR scenarios because it considers the high noise cases as collisions. The classifier accuracy and error performance with respect to SNR are shown in Fig. 10.

C. Full Algorithm Performance

The main objective for implementing the reconstruction algorithm is to save time when reconstructing the signal.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

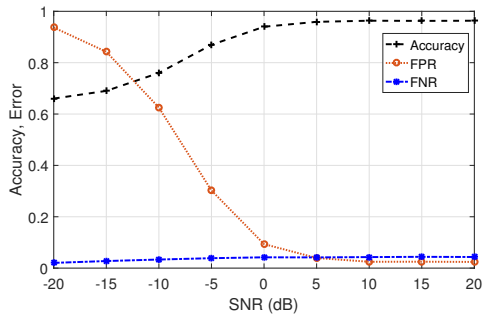


Fig. 10. Collision Classifier accuracy and errors.

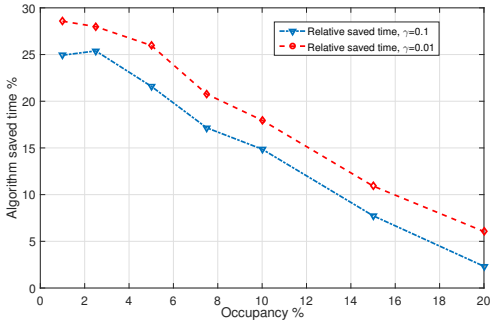


Fig. 11. Time saving relative to two variables solution, SNR=0 dB.

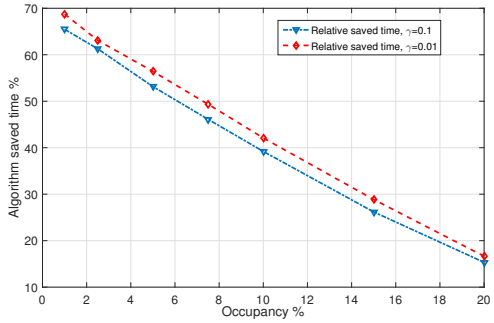


Fig. 12. Time saving relative to three variables solution, SNR=0 dB.

Hence, we evaluate time saving relative to a fixed variable resolver. Two cases are considered for comparison. In the first case, the solution of fixed $v_\ell = 2$ variables is compared to the reconstruction algorithm in which v_ℓ is set to 2 when a collision is detected, and $v_\ell = 1$ otherwise. Fig. 11 shows the time saved by implementing the algorithm. Similarly, Fig. 12 shows the relative saved time by implementing the reconstruction algorithm for $v_\ell = 3$. Obviously, the time saved for the second case is much higher due to the fact that for $v_\ell = 3$ and $D = 10$, 120 combinations of three variables, compared to 45 combinations of two variables for $v_\ell = 2$ and $D = 10$, are solved for each sub-Nyquist bin. The saved time is the result of solving for only 10 one variable combinations instead, whenever no collision occurs. In both cases, the saved time is inversely proportional to the occupancy level because of a lower number of collisions at low occupancy levels. Furthermore, the saved time is higher for lower threshold because more bins are taken into consideration. While a trade-off between saved time and detection accuracy exists, Table I shows that the change in P_D and P_{FA} values as a result

TABLE I
EFFECT OF COLLISION DETECTOR ON P_D AND P_{FA}, SNR=0 dB

		$v_\ell = 2$, $\gamma = 0.1$	$v_\ell = 2$, $\gamma = 0.01$	$v_\ell = 3$, $\gamma = 0.1$	$v_\ell = 3$, $\gamma = 0.01$
P_D decrease	Max	0.0178	0.0134	0.0249	0.0227
	Mean	0.0112	0.0069	0.0168	0.0160
P_{FA} decrease	Max	0.0465	0.0537	0.1182	0.1285
	Mean	0.0333	0.0399	0.0794	0.0931

of using the collision detector is insignificant. The first and second columns show the maximum and the mean of decrease in P_D and P_{FA} values for Fig. 11. The third and fourth columns show the same values for Fig. 12.

V. CONCLUDING REMARKS

In this paper, we studied a sparse spectrum reconstruction method in which a delay-based multicoset system is employed to reconstruct a sub-sampled signal. While many other sub-Nyquist systems exist, this system is preferred for practical implementation due to its simplicity and relatively fast signal reconstruction time. We defined the general reconstruction model and derived the multi-combination least square approach. In general, solving for a higher order of variables even when there are less active tones in a sub-Nyquist bin produces more accurate estimates. However, higher order solutions require more time in addition to more branches.

In order to reduce the time needed to reconstruct the signal, we proposed a classifier to classify the aliased sub-Nyquist bins. While the total accuracy of the classifier decreases as SNR decreases, the reduction in the accuracy has minimum effect on the overall detection performance. However, more time is gained for more accurate classification. The saved time for the algorithm resolving a maximum of three variables is more than double that of algorithm resolving up to two variables. This is crucial since detection performance of the three-variable resolver is much higher than that of the two-variable resolver, provided that there are enough branches to solve for three variables.

REFERENCES

[1] A. Nallanathan, "Wideband spectrum sensing for cognitive radio networks: a survey," *IEEE Wirel. Commun.*, vol. 20, no. 2, pp. 74–81, 2013.
[2] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 375–391, April 2010.
[3] D. D. Arianandа and G. Leus, "Compressive wideband power spectrum estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4775–4789, Sept 2012.
[4] P. P. Vaidyanathan and P. Pal, "Sparse sensing with co-prime samplers and arrays," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 573–586, Feb 2011.
[5] F. A. Sakarya, G. S. Nagel, L. H. Tran, J. a. Molnar, and F. Ayhan Sakarya, "Wideband compressed sensing for cognitive radios," *Proc. IEEE Milcom*, pp. 31–36, 2011.
[6] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "GHz-wide sensing and decoding using the sparse Fourier transform," *Proc. IEEE INFOCOM*, pp. 2256–2264, 2014.
[7] J. Demmel, *Applied Numerical Linear Algebra*, 1st ed. Society for Industrial and Applied Mathematics, 1997.
[8] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

A Tetherless, Absolute-Time Channel Sounder; Processing and Results for a Complex Environment*

David Novotny[1], Alexandra Curtin[2], Jeanne Quimby[2], Kate Remley[2], Peter Papazian[2], Richard Candell[3]
[1] National Institute of Standards and Technology: Communications Technology Laboratory, Boulder, CO, USA, david.novotny@nist.gov
[2] National Institute of Standards and Technology: Communications Technology Laboratory, Boulder, CO, USA
[3] National Institute of Standards and Technology: Engineering Laboratory, Gaithersburg, MD, USA

***Abstract*— We present a channel sounder that can operate without a tether and still maintain an absolute time reference between the source and receiver. Based on a sliding correlator, with synchronized rubidium clocks to generate phase references for the up- and down- converted RF carriers, and a synchronous trigger, the system generates locked signals in the short term (tens of hours). The system has an operational range of 10 MHz to 6 GHz with an instantaneous channel bandwidth of up to 200 MHz.**

We start with a discussion on processing measurements for oversampled band-limited signals. Spectral truncation is compared with transmit spectrum filtering; DC bias removal and referencing to remove systematic effects are discussed.

We conclude with channel sounding results, power delay profile, RMS delay spread, and time of arrival versus position for an electromagnetically complex environment.

***Index Terms*— channel sounding, propagation, power delay profile, impulse response.**

I. INTRODUCTION

The realized propagation path between two points can vary greatly from Friis transmission in non-free-space environments. Basic free space loss [1] does not adequately explain the multi-path and lossy environments seen by modern communications systems. The ability to estimate the propagation characteristics between multiple points impacts radio channel quality, radio system capacity and thus radio system design and, more importantly, cost.

Many channel sounders have been built over different frequency ranges [2-11]. They have used single frequency, wideband noise and patterns that emulate protocol and data load to sound the channel. One major hurdle is that the propagation channel is not static, but can change rapidly and vary greatly in loss and dispersion. Characterizing many environments to get the spread of the data and losses is an arduous task. Comparing data from many measurements using different methods is difficult. The decision on what data to use, what scenarios need to be addressed, and how to cover the greatest range of realistic propagation environments is diffcult.

Many sounding analyses report the channel (TX antenna, propagation environment and RX antenna) convolved with a transmit spectrum of the sounder. To use the sounding results for a general transmit spectrum, the original transmit spectrum can be removed and the desired transmit spectrum can be re-applied. We propose a slightly different method for data reporting. By reporting only a channel response to frequencies with high signal to noise ratio (SNR), the response through a system can be estimated by convolving the reported response with an arbitrary transmit sequence.

We present a tetherless absolute-time channel sounder and methods for processing the results to represent the response of a specific data channel with limited dependence on the transmit spectrum of the sounder. These measurements were done in manufacturing facilities to help investigate the penetration of wireless networking signals in a very multi-path rich environment. Data were taken at three manufacturing facilities near two Industrial, Medical and Scientific (ISM) bands at 2.45 and 5.8 GHz. This paper only addresses part of the data; however, all of the data taken are available to independent third parties to perform comparative analysis [12].

II. SYSTEM METHODLOGY

Two major design decisions drove the overall architecture of our channel sounder: tetherless operation and absolute time referencing. Tetherless operation allows not only for large physical distance variations, but the ability to measure complex operational environments with minimal disturbance to the environment itself. The absolute time reference provides detailed delay information for the channel. The ability to measure absolute delay without Global Positioning Systems (GPS) allows for accurate channel measurements in cluttered and shadowed areas. The single step referencing performed here does require physical connection between the source and receiver but in return, it removes all (non-channel) linear systematic delays and losses.

A. System Architecture

Fig. 1 shows an overall block diagram of the system. A recurring, oversampled pseudorandom (PN) code word is triggered repetitively and transmitted through an amplifier and a matched filter to limit radiated harmonics. The amplified signal is sampled to determine the actual power transmitted. Finally the signal is routed through an attenuator for referencing or the antennas to determine a complex impulse response (CIR) or power delay profile (PDP).

* US Government Work, Not Subject to US Copyright.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Two stable and independent timing chains maintain synchronization between the separated transmitter and receiver. Trigger timing and the radio frequency (RF) up- and down-conversion must be locked and synchronized. Rubidium (Rb) clocks have been used extensively in the channel sounding community to synchronize local oscillators (LO) for frequency conversion [10-11]. We use these clocks in the frequency conversion process. Additionally, a synchronized timing reference is generated with the pulse per-second (PPS) output from the clock. This Rb-sourced PPS signal is divided by the synchronization hardware to create a reference trigger to coherently initiate signal generation and acquisition. The redundancy of 10 MHz signals to the triggering and frequency conversion sections of the system (see Fig. 2) limits the amount of distribution jitter to minimize phase and time drift.

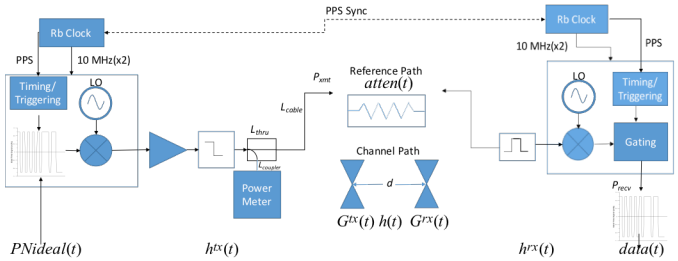


Figure 1. System architecture of the channel sounder. Measuring through the reference attenuator generates a known loss and delay reference that can be compared to the channel data through the antennas and propagation path to determine the channel complex impulse response.

B. *Calibration Steps*

- Transmit losses L_{thru}, $L_{coupler}$, L_{cable}, $L_{attenuator}$ are measured to allow for accurate calculation of the actual transmitted power and reference loss.
- Vector signal transceiver (VST) is power calibrated and referenced against an external power meter.
- PPS is linked between Rb clocks and allowed to stabilize.
- PPS is reestablished between the transmitter and receiver. Each chassis is disciplined to the PPS from its clock. This creates an accurate timing frame between the two chassis with the nominal stability of the clocks ($1x10^{-11}$ sec/100 sec).
- A synchronized trigger is generated relative to the PPS.
- A reference measurement is taken through the attenuator to establish delay and loss between the chassis.

III. Calculations of Channel Impulse Response and Power Delay Profile.

A. *Removal of Systematic Components.*

From Fig. 1, the output data, *data(t)*, is the input PN code word, $PN_{ideal}(t)$, transmitted through the transmitting system, $h^{tx}(t)$, the channel represented by the transmitting antenna, $G^{tx}(t)$, the environment, $h(t)$, and the receiving antenna, $G^{rx}(t)$, and finally the receiving system, $h^{rx}(t)$. During the reference calibration, the channel, $G^{tx}(t)*h(t)*G^{rx}(t)$, is replaced by an attenuator, *atten(t)*, so the systematic effects of the measurement system can be minimized.

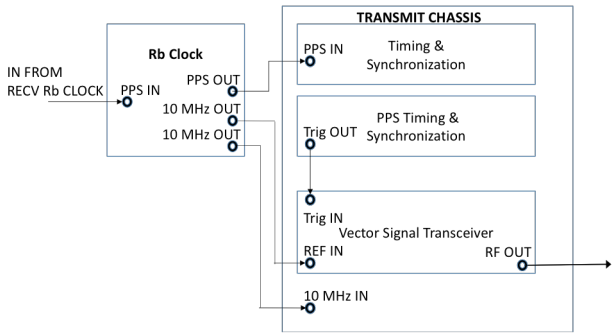


Figure 2. Timing and synchronization connections for the channel sounder. The multiple connections between the clock and chassis limit 10 MHz propagation errors between chassis components. The 10 MHz reference to the chassis provides inter-function synchronization while the 10 MHz to the transceiver provides a reference for frequency conversion. The receive chassis mirrors this setup.

The received data can be expressed equivalently in the time and frequency domains:

$$\begin{aligned} &data(t) = PN_{ideal}(t) * h^{tx}(t) * G^{tx}(t) * h(t) * G^{rx}(t) * h^{rx}(t), \\ &\boldsymbol{data}(f) = \boldsymbol{\mathcal{F}}(data(t)) = \\ &\boldsymbol{PN}_{ideal}(f) \cdot \boldsymbol{h}^{tx}(f) \cdot \boldsymbol{G}^{tx}(f) \cdot \boldsymbol{h}(f) \cdot \boldsymbol{G}^{rx}(f) \cdot \boldsymbol{h}^{rx}(f), \end{aligned} \quad (1)$$

where $\boldsymbol{\mathcal{F}}$ denotes the Fourier transform, the "*" operator denotes convolution in the time domain, and the "•" operator denotes frequency-by-frequency multiplication. The channel's desired complex impulse response, *CIR(t)*, is given by just the radiated portion of the measurement:

$$CIR(t) = G^{tx}(t) * h(t) * G^{rx}(t). \quad (2)$$

A reference measurement is taken through a known attenuator. The resultant measurement yields a nominal characterization of the measurement system without the channel:

$$\begin{aligned} &ref(t) = PN_{ideal}(t) * h^{tx}(t) * atten(t) * h^{rx}(t), \\ &\boldsymbol{ref}(f) = \boldsymbol{\mathcal{F}}(ref(t)) = \boldsymbol{PN}_{ideal}(f) \cdot \boldsymbol{h}^{tx}(f) \cdot \boldsymbol{atten}(f) \cdot \boldsymbol{h}^{rx}(f). \end{aligned} \quad (3)$$

Assuming the system is linear with received power and time stable, this allows for a normalization of the measurement by the reference to yield *CIR(f)*:

$$\begin{aligned} &\frac{\boldsymbol{data}(f)}{\boldsymbol{ref}(f)} = \\ &\frac{\boldsymbol{PN}_{ideal}(f) \cdot \boldsymbol{h}^{tx}(f) \cdot \boldsymbol{G}^{tx}(f) \cdot \boldsymbol{h}(f) \cdot \boldsymbol{G}^{rx}(f) \cdot \boldsymbol{h}^{rx}(f)}{\boldsymbol{PN}_{ideal}(f) \cdot \boldsymbol{h}^{tx}(f) \cdot \boldsymbol{atten}(f) \cdot \boldsymbol{h}^{rx}(f)} = \\ &\frac{\boldsymbol{CIR}(f)}{\boldsymbol{atten}(f)}. \end{aligned} \quad (4)$$

This can be rewritten into the time domain:

$$CIR(t) = \boldsymbol{\mathcal{F}}^{-1}\left[\boldsymbol{CIR}(f)\right] = \boldsymbol{\mathcal{F}}^{-1}\left[\frac{\boldsymbol{\mathcal{F}}\left[data(t)\right]}{\boldsymbol{\mathcal{F}}\left[ref(t)\right]} \cdot \boldsymbol{atten}(f)\right]. \quad (5)$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

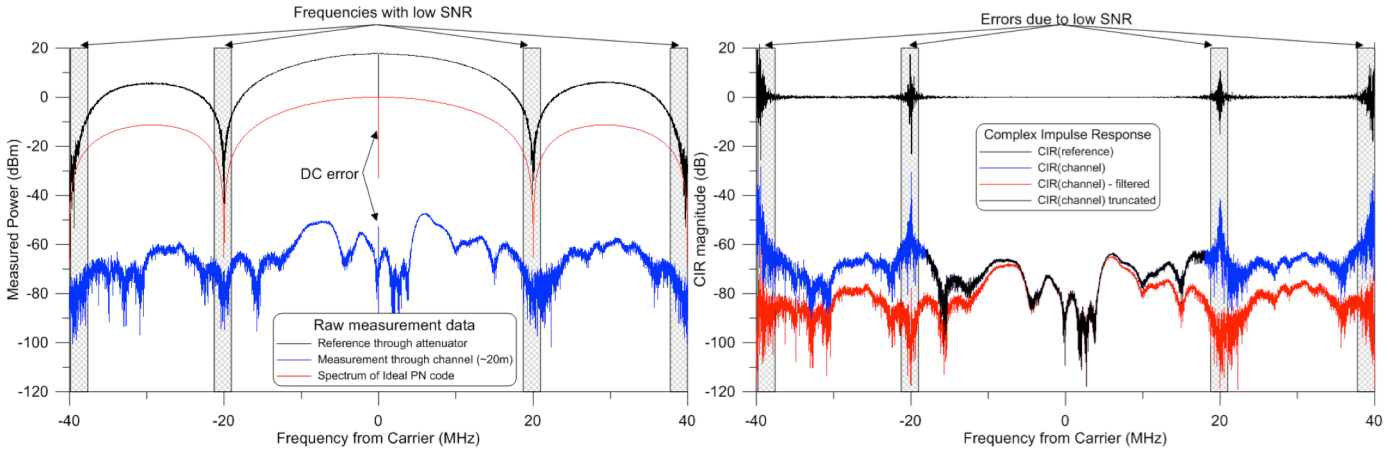


Figure 3. Spectrum of the reference and a measurement (left) and the CIR of the reference and measurement (right) show the effects of reference normalization and filtering options. DC bias removal, filtering, and proper truncation limit transitions that raise the noise floor in the PDP.

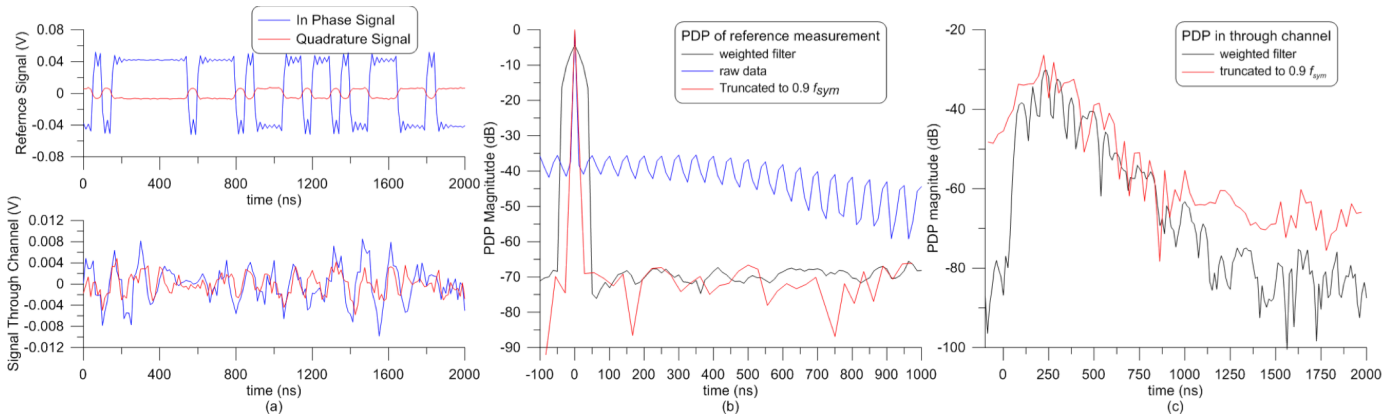


Figure 4. (a) Raw time-domain reference and measurement signals; tracking the I/Q phase provides Doppler information for rapidly changing channels. The PDP of the reference (b) shows the desired impulse at zero time, the unfiltered, raw data shows ringing due to the errors shown in Fig.3. Truncation (c) has a larger time step and the step at the end of the frequency record does show a higher noise floor than the weighted filter.

Note that the *atten(f)* is the transmission coefficient of the attenuator, so a frequency invariant 50 dB attenuator will have $atten(t) = atten(f) = 10^{-50/20} \approx 0.00316$.

B. *Systematic Error Due to DC Biasing.*

DC sampling errors can come from two sources: down-conversion errors and sampler offsets. The down-conversion errors primarily arise from small frequency errors between the transmitter and receiver, and from mixer leakage. At low signal levels, the DC bias in the sampler may be a significant source of error especially once transformed back to the time domain. The net DC error is often seen as a transition right at the carrier frequency or zero frequency in the down converted spectrum, (see Fig. 3). Practically, DC can't be transmitted over the air, and the DC error is compressed into one frequency component. Finally, the frequency components in the measurements are correlated through the FFT. A simple correction for fixing the DC error can be performed through a complex average of the points on either side of the DC term [13]:

$$CIR(0) = \frac{CIR(\Delta f) + CIR(-\Delta f)}{2}. \qquad (6)$$

The PDP can be generated from the corrected *CIR(t)*:

$$PDP(t) = \left|CIR(t)\right|^2. \qquad (7)$$

C. *Addressing Spectral Power, Noise and Over Sampling.*

The calculations of the CIR and PDP in (6) and (7) require a division by the spectrum of the reference through the attenuator. A transmitter has a limited bandwidth (limited by allowable TX power, regulation, or frequency of interest). Further, in order to improve dynamic range performance, the received signal is often oversampled in time to improve correlation results. This oversampling extends the effective measured frequency range. However, frequencies above the symbol frequency lack spectral power. This results in some measured frequencies with little transmitted power resulting in the noise/noise issue in (5), see Fig. 3. While using a BPSK PN code offers considerable processing gain, it lacks spectral strength above the symbol frequency f_{sym}. Other modulations and filters (matched cosine) can be used, but all over-sampled time-domain based systems will suffer low SNR over some portion of the measured frequency spectrum.

1) *Filtering the Complex Impulse Response.*

A common method to normalize energy lost to the filtering process is to correct for the energy of the applied filter [14]. This filtering creates an effective CIR:

$$CIR_{eff}(t) = w'(t) * CIR(t) = \frac{w(t)}{\sqrt{\int_t w(t)\,dt}} * CIR(t). \qquad (8)$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Various filters can be used for *w(t)*; the primary requirement is to have a frequency zero at the symbol frequency. Brick-wall filters can be effective, but may introduce time-domain ringing. A common implementation is to use the ideal spectral power of the transmitted signal (see Fig. 3) as the filter, *w(t)*. This filter has an ideal zero power null that coincides with the nulls in the transmit spectrum. It does reduce the peak level of the ideal impulse and spreads it out in time. This can result in the measured path loss and the root mean square (RMS) delay spread, both critical communications parameters, being a function of the applied filter versus a physical characteristic of the channel itself.

2) Frequency Truncation of Complex Impulse Response.

Another filtering approach is to truncate the frequency range of the measurement to the frequencies with high SNR. This is different than applying a brick wall filter. In this case, the out of band components are removed, not zero filled. This limits the change in the PDP due to signal processing while maintaining the processing gain of the long over-sampled sequence, but at the price of reduced effective sampling and temporal resolution in the final PDP.

A suggested frequency truncation window is to limit the frequency extent of the CIR to where the transmitted signal drops below a given level. For a *sin(f)/f* spectrum of a PN code, the -20 dB level correlates to approximately $\pm 0.9\, f_{sym}$, and -30 dB uses approximately $\pm 0.97\, f_{sym}$. Practically, the filtering methods used in [13-15] limit the utilized spectrum to approximately the 20 to 30 dB SNR level, but a truncation approach reduces variability due to the chosen filter.

3) Comparitive Discussions of Filtering.

Truncation limits the reported frequency coverage and time-domain resolution. However, it reports lower uncertainties in the declared frequency channel. For this case of a 20-MHz symbol rate, a ±20-MHz or 40-MHz channel response is often reported. However, the uncertainty is higher near the band edges. Truncating to the -20 dB level, in this case ±18 MHz or 36 MHz, returns results with smaller noise related uncertainties in the reported band, however Fig. 4 also shows the potential for a higher noise floor and possibly less dynamic range due to reducing the processing gain from lower oversampling.

Optimizations can be made; using root-raised-cosine pulse shaping can flatten the transmitted spectrum and limit nulls. However, oversampling will still result in frequency regions of little power. Post-normalization filtering or truncation is needed to reduce the frequencies to bands of interest.

IV. Channel PDP Data From Two Manufacturing Environments.

The purpose of channel sounders is to measure how signals propagate in real environments. Measurements using the channel sounder described here were performed at two different factory environments. These measurements were focused on determining optimal placement of wireless IEEE 802.11 infrastructure, as well as the basic operational validity of the sounder. To emulate the propagation characteristics of the 802.11 bands, but yet not interfere with nearby installed systems, tests were performed in government bands at 2.245 GHz and 5.4 GHz. Table I shows the measurement parameters used. While all the data are available online for multiple facilities, we present data for one transmitter position (TX1) and two receiver polarizations at 2.245 GHz. We present a CIR, PDP, and RMS delay spread for a truncated channel. Results for a traditional PN channel sounder can be generated by convolving the reported CIR [12] with a PN spectrum [14].

A. Measurement Parameters

TABLE I. Measuremnt Parameters

Center Frequency	2.245 GHz (emulates 2.4 GHz 802.11 b/g/n) 5.412 GHz (emulates 5 GHz 802.11 a/n/ac)
Transmit Power	2.245 GHz – 1.5W 5.400 GHz – 1.25W
Dynamic Range	130 dB insertion loss
Bandwidth	40 MHz (null to null)
PN code length	2047 symbols
Transmit sample rate	Effective symbol rate: 20 MS/s 2x oversampling = 80 MHz sample rate
Receive sample rate	80 MS/s
Effective codeword length	2047 symbols · 2 samples/bit · 2x oversample · 12.5ns/bit = 102.350 µs
Data save rate	Every 200 code words = 20.47 ms Effectively 4.5 MB/sec on disk
Tx antenna	Vertically polarized bi-conical 2.9 dBi max gain @ 2.245 GHz 3.6 dBi max gain @ 5.412 GHz
Rx antenna	Broadband dipole – three orthogonal polarizations -4.2 dBi max gain @ 2.245 GHz -3.5 dBi max gain @ 5.412 GHz
Tx height	5 m
Rx height	1.5 m – 2 m

B. Measurement Path

We present results from a machine shop floor at NIST in Gaithersburg, MD fig. 6. This facility was used as a small-scale version of a commercial industrial facility. It has a large and cluttered ~ 40 x 20 m open area with a ceiling height of 8 m. The walls are concrete block, the floor is reinforced concrete, and the ceiling is steel. There are no exterior windows. Pictures the measurement system are shown in Fig 5.

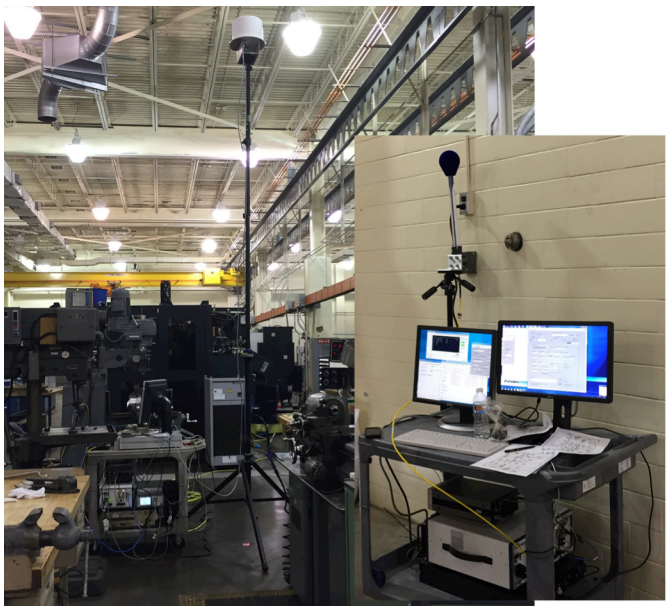


Figure 5. Picture of transmit setup at TX2 (left) and receiver (right).

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

C. PDP Measuremnt

The PDPs for the transmitter at TX1 and vertically and horizontally polarized receive antennas are calculated using a truncation at 90% of the symbol rate (±~18 MHz channel centered on 2.245 GHz) with no filtering, Fig. 7. The mismatch in polarization results in an approximate 10 dB excess loss compared to the co-polarized case and a lengthened PDP.

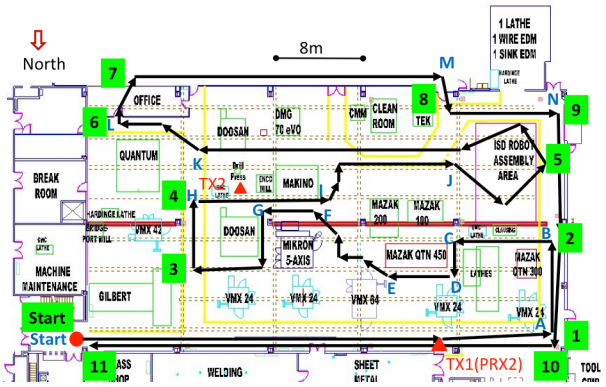


Figure 6. A map of the facility and the measurement path where channel sounding measurements were taken. While code word transmission was continuous, soundings were saved every 20 ms or at ~2 cm intervals.

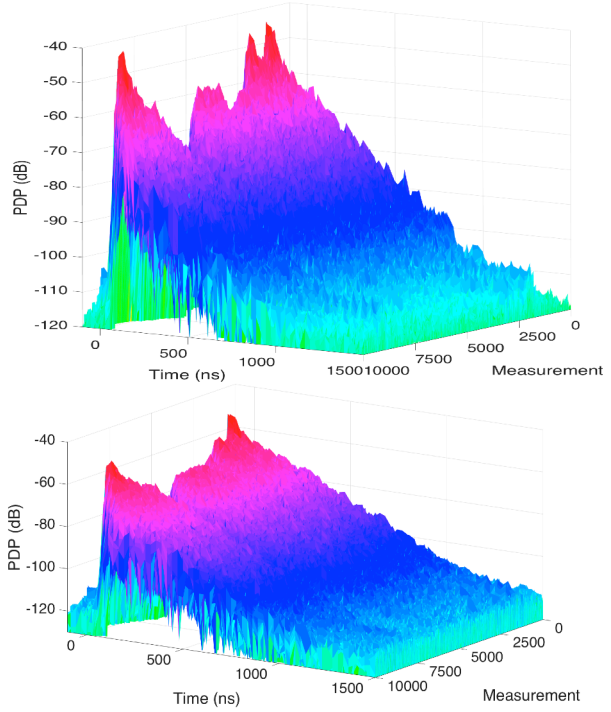


Figure 7. PDP from source location TX1 to a vertically (top) and horizontally (bottom) polarized receive antenna as it transits along the path in Fig. 6. There is a nominal 8-10 dB greater loss for the cross-polarized horizontal case.

D. Path Loss and RMS Delay Spread

The two major parameters generated from the PDP are path loss and RMS delay spread [17]. The path loss is a measure of the propagation loss that needs to be overcome by a combination of antenna gain, transmit power, and receiver sensitivity. The RMS delay spread is a measure of the delay dispersion of the channel (free space= 0 ns) which, in practice, limits the maximum symbol rate in the measured channel due to multi-path induced inter-symbol interference.

The path loss compared to ideal free space for both polarizations is shown in Fig. 8.

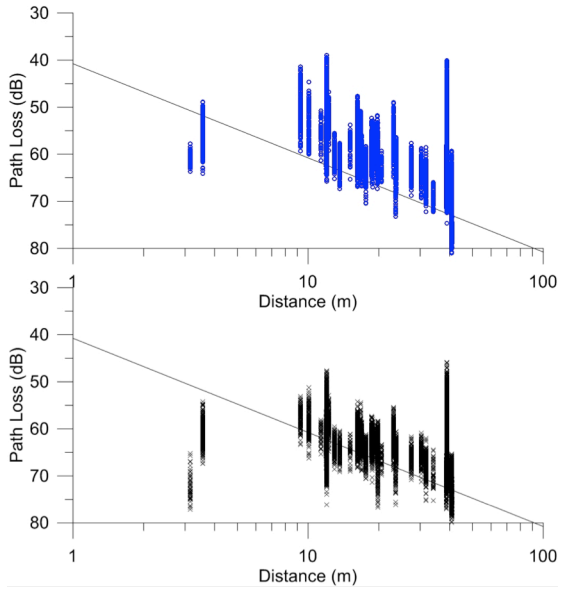


Figure 8. Integrated path loss for both polarizations shown in Fig. 7. The co-polarized transmitter and receiver (vertical/top) show approximately 10 dB less path loss than the cross-polarized (horizontal/bottom) case.

The RMS delay spread with a -20 dB threshold [17] shows similar degradation in the cross-polarized case, Fig. 9.

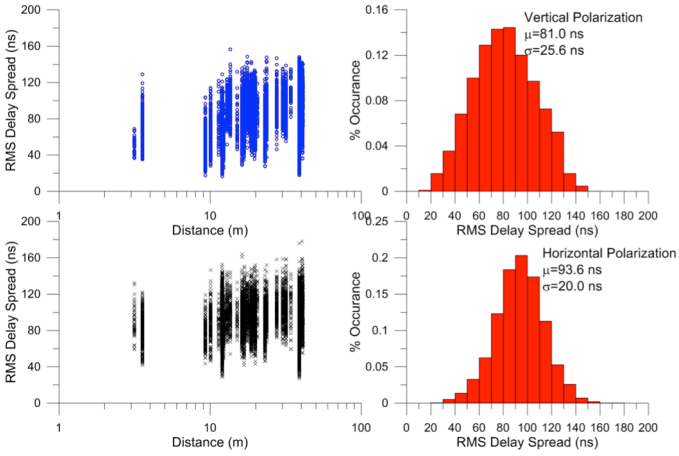


Figure 9. RMS delay spread for two polarizations.

E. Time of Flight and First Arrival.

The channel sounder presented has the ability to measure absolute time between the transmitter and receiver. This can be compared to first peak and maximum peak arrival times, Figs. 10,11. This information can be used to infer line-of-sight suppression, reverberation energy [16], and mixing ratio.

Data in Figs. 10,11 highlight a challenge of wireless communication in an electromagnetically (EM) complex environment. Multi-path propagation can be subject to inter-symbol interference when multi-path is comparable or higher than the direct signal. Proper direction analysis in MIMO systems may reduce errors and improve link quality.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

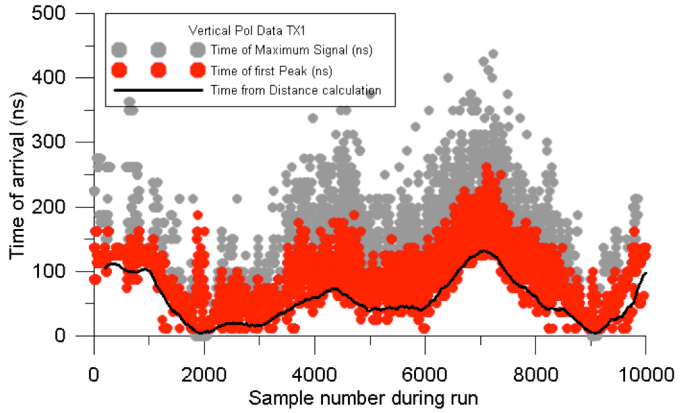


Figure 10. Time of first arrival, time of maximum signal, and distance calculation for the vertically polarized receiver along the path of Fig. 6. The time maximum and first arrival signal do not always correspond to predicted line-of-sight time-of-flight.

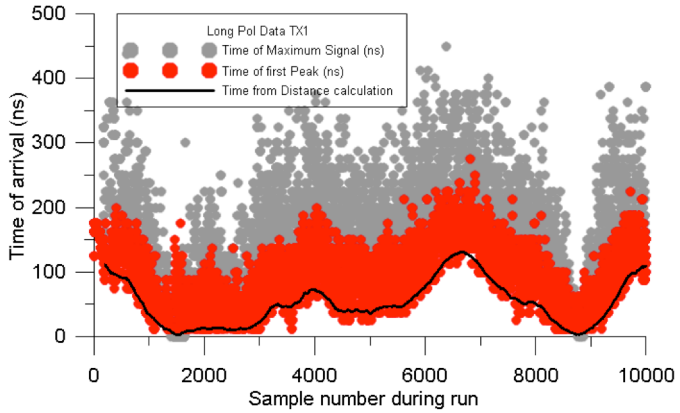


Figure 11. Time of first arrival, time of maximum signal, and distance calculation for the horizontally polarized receiver along the path of Fig. 6.

V. CONCLUSIONS

We have presented a 10 MHz-6 GHz channel sounder with absolute time measurement capability. We have covered methods for processing and filtering the data to account for low signal-to-noise caused by band-edge and oversampling. A frequency truncation proposal for determining PDP and derived parameters using frequencies that only have high SNR and has direct correlation to the reported band of significance is presented.

Data for several EM-cluttered facilities were taken and are available for processing by interested parties [12]. Processed data for a small subset were presented to show the results of frequency truncation, antenna polarization, and absolute time capabilities of the new sounder.

REFERENCES

[1] H.T. Friis, "A note on a simple transmission formula," in Proceedings of the IRE , vol.34, no.5, pp.254-256, May 1946.

[2] J. D Parsons, D. A. Demery, A. M. D. Turkmanil, "Sounding techniques for wideband mobile radio channels: a review," IEE Processings communication, speech and vision., Vol. 138, No. 5, 437-446, 1991.

[3] T. Zwick, T. J. Beukema, Haewoon Nam, "Wideband channel sounder with measurements and model for the 60 GHz indoor radio channel," in Vehicular Technology, IEEE Transactions on , vol.54, no.4, pp.1266-1277, July 2005.

[4] V. Kolmonen, P. Almers, J. Salmi, J. Koivunen, K. Haneda, A. Richter, F. Tufvesson, A.F. Molisch, P. Vainikainen, "A dynamic dual-link wideband MIMO channel sounder for 5.3 GHz," in Instrumentation and Measurement, IEEE Transactions on , vol.59, no.4, pp.873-883, April 2010.

[5] J. Peigang, W. Shaobo; L. Huajia, "An effective solution of wireless channel sounder and its channel modeling application," in Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th , vol.1, no., pp.249-253 Vol.1, 17-19 May 2004.

[6] K. Mahler, P. Paschalidis, A. Kortke, M. Peter, W. Keusgen, "Realistic IEEE 802.11p transmission simulations based on channel sounder measurement data," in Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th , vol., no., pp.1-5, 2-5 Sept. 2013.

[7] X. H. Mao, Y. H. Lee, B. C. Ng, "Wideband channel modelling in UHF band for urban area," IEEE International Conference on Wireless Communication Systems, 240-244, Reykjavik, Iceland, October 2008.

[8] T.S. Rappaport, G. R. MacCartney, M.K. Samimi, Shu Sun, "Wideband Millimeter-Wave Propagation Measurements and Channel Models for Future Wireless Communication System Design," in Communications, IEEE Transactions on , vol.63, no.9, pp.3029-3056, Sept. 2015.

[9] Y. Azar, G.N. Wong, K. Wang, R. Mayzus, J.K. Schulz, Hang Zhao, F. Gutierrez, D. Hwang, T.S. Rappaport, "28 GHz propagation measurements for outdoor cellular communications using steerable beam antennas in New York city," in Communications (ICC), 2013 IEEE International Conference on , vol., no., pp.5143-5147, 9-13 June 2013.

[10] N. Yoza, K. Baker, "Narrowband 5 GHz Mobile Channel Characterization," Proc of 37th Meeting of the Antenna and Measurements Techniques Assoc., pp. 335-340, October 2015.

[11] K. Sarabandi, N. Behdad, A. Nashashibi, M. Casciato, L. Pierce, F.Wang, "A measurement system for ultrawide-band communication channel characterization," IEEE Trans. Antennas Propag., vol. 53, pp. 2146-2155, July 2005.

[12] http://www.nist.gov/el/isd/cs/wsie_data.cfm

[13] Richard G. Lyons, Understanding Digital Signal Processing: International Edition, 3rd ed, Prentice Hall, 2011.

[14] R. J. Pirkl and G. D. Durgin, "Revisiting the spread spectrum sliding correlator: why filtering matters," in *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp. 3454-3457, July 2009. doi: 10.1109/TWC.2009.081388

[15] P. B. Papazian, J. Choi, J. Senic, P. Jeavons, C. Gentille, N. Golmie, R. Sun, D. Novotny, K.A. Remley, "Calibration of millimeter-wave channel sounders for super-resolution multipath component extraction," *2016 10th European Conference on Antennas and Propagation (EuCAP)*, Davos, 2016, pp. 1-5.

[16] H. Fielitz, K.A. Remley, C.L. Holloway, Qian Zhang, Qiong Wu, D.W. Matolak, "Reverberation-chamber test environment for outdoor urban wireless propagation studies," in Antennas and Wireless Propagation Letters, IEEE , vol.9, no., pp.52-56, 2010.

[17] "Recommendation ITU-R P.1407-5 (09/2013)-Multipath propagation and parameterization of its characteristics," Internatoinal Telecommunications Union, Sept, 2013, https://www.itu.int/rec/R-REC-P.1407-5-201309-I/en.

Group Discovery Time in Device-to-Device (D2D) Proximity Services (ProSe) Networks

David Griffith, Aziza Ben Mosbah, and Richard Rouil
National Institute of Standards and Technology
Gaithersburg, Maryland 20899–8920
Email: david.griffith@nist.gov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

***Abstract*—Device-to-device (D2D) communications for Long Term Evolution (LTE) networks relies on a discovery process to enable User Equipment (UE) to determine which D2D applications and services are supported by neighboring UEs. This is especially important for groups of UEs that operate outside the coverage area of any base station. The amount of time required for discovery information to reach every UE in a group depends on the number of UEs in the group and the dimensions of the discovery resource pool associated with the Physical Sidelink Discovery Channel (PSDCH); an additional factor is the half-duplex property of current UEs. In this paper, we use a Markov chain to characterize the performance of Mode 2 direct discovery. The resulting analytical model gives the distribution of the time for a UE to discover all other UEs in its group. We validate the model using Monte Carlo and network simulations.**

I. Introduction

Proximity Services (ProSe) for Long Term Evolution (LTE) was developed by 3rd Generation Partnership Project (3GPP) for Device-to-Device (D2D) communications. The standard will allow User Equipments (UEs) to communicate directly with other UEs that are within range by using a portion of the channel known as the sidelink (SL). ProSe covers UEs that are in the coverage area of an evolved Node-B (eNB), in which case the eNB can coordinate SL resource allocation, but it also supports communication between UEs that are out-of-coverage with respect to any eNB. This affects public safety applications, which may involve deployment of personnel to remote areas, or deployment in disaster areas where the infrastructure has been destroyed.

UEs use discovery messages to exchange information regarding their D2D applications and capabilities. Discovery messages use the Physical Sidelink Discovery CHannel (PSDCH). In this paper, we consider Mode 2 discovery, where PSDCH resources are not allocated to individual UEs but are available for all UEs to use; this mode applies to the out-of-coverage case. UEs pick PSDCH resources randomly, which creates the risk of message loss due to collisions when more than one UE selects a given resource. Allocating more resources to the PSDCH reduces the collision rate, but it also reduces the bandwidth for data transmission. In order to properly size the discovery resource pool and to determine the maximum size of a group of D2D UEs that the network can support, it is important to develop models that allow operators to accurately measure the performance of the PSDCH.

This work extends our previous study of the PSDCH [1], which used an analytical model to get the discovery message transmission probability that maximizes the message success probability between two UEs in a single period. Sarret et al. used simulations to get the time for a UE to be discovered by all other UEs in a group, considering both half-duplex and full duplex UEs [2]. Lin et al. examined two D2D discovery schemes and used a Markov chain to model a backoff procedure for an individual UE [3]. Zhang and Liu examined hopping patterns in the discovery resource pool for half-duplex UEs and used simulations to compare their performance with respect to the discovery rate and cumulative number of discovered UEs [4].

In this paper, we develop a Markov chain-based analytical model that gives the distribution of the time for a single UE in a D2D group to discover all other UEs in the group. First, we describe the Physical Sidelink Discovery Channel (PSDCH) and the assumptions behind our model in Section II. In Section III, we derive an analytical expression for the elements of the Markov chain's state transition probability matrix, which we use to obtain the cumulative distribution function (CDF) for the group discovery time. In Section IV, we validate the theoretical model from Section III using both Monte Carlo simulations in Matlab and network simulations in NS3, and we demonstrate the model's use by obtaining the maximum UE group size that allows a UE to discover all of its peers within a required number of periods with a given probability. We discuss extensions for the model and summarize our work in Section V. In Table I, we provide a list of the symbols that we use in this paper.

II. Modeling the Discovery Resource Pool

In this section, we describe D2D discovery message transmission and discuss the assumptions that underlie the analysis in Section III. We assume that UEs transmit discovery messages during every occurrence of the PSDCH discovery resource pool, which repeats periodically with period P [5, Clause 14.3.3], as described in [6, Clause 8.3]. A pool resource

Disclaimer: Certain commercial software packages are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the software packages identified are necessarily the best available for the purpose.

Ben Mosbah, Aziza; Griffith, David; Rouil, Richard.
"Group Discovery Time in Device-to-Device (D2D) Proximity Services (ProSe) Networks."
Paper presented at IEEE INFOCOM 2017 - The 36th Annual IEEE International Conference on Computer Communications,
Atlanta, GA. May 1, 2017 - May 4, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

is a single transport block, which is composed of a pair of adjacent Physical Resource Blocks (PRBs) that occupy the same subframe [5, Clause 14.3.1.1]. Discovery messages can be repeated in a given period up to four times (i.e., one initial transmission followed by zero, one, two, or three retransmissions); the number of transmissions is given by the parameter N_{SLD}^{TX} [5, Clause 14.3.1.1].

Increasing N_{SLD}^{TX} decreases the number of available resources in the pool, since the resource pool's effective dimensions in the frequency and time domains are respectively $N_f = \lfloor M_{RB}^{PSDCH\_RP}/2 \rfloor$ and $N_t = \lfloor L_{PSDCH}/N_{SLD}^{TX} \rfloor$, where $M_{RB}^{PSDCH\_RP}$ is the number of PRBs in the pool and L_{PSDCH} is the number of subframes spanned by the pool in the time domain. The total number of resources in the pool is thus $N_r = N_f N_t$.

We assume that all UEs are half-duplex. We also assume that the UEs use Mode 2 discovery [5, Clause 14.3.1.1]. In each period, every UE, independently of every other UE, generates a uniformly random resource index $n_{PSDCH} \in \{0, 1, \ldots, N_r - 1\}$, which maps to a unique set of PRB and subframe indices via the following equations:

$$a_j^{(i)} = \left((j-1)\lfloor N_f / N_{SLD}^{TX} \rfloor + \lfloor n_{PSDCH}/N_t \rfloor\right) \bmod N_f \quad (1a)$$

$$b_j^{(i)} = n_{PSDCH} \bmod N_t, \quad (1b)$$

where i is the period index and $1 \le j \le N_{SLD}^{TX}$, so that j indexes the transmission attempts in the ith period. The parameters $a_j^{(i)}$ and $b_j^{(i)}$ map to the subframe and PRB pair indices $l^{PSDCH}_{N_{SLD}^{TX} b_j^{(i)}+j-1}$ and $\left(m^{PSDCH}_{2a_j^{(i)}}, m^{PSDCH}_{2a_j^{(i)}+1}\right)$, respectively. A UE can throttle its message transmissions by generating a $[0, 1]$-uniform random variate and transmitting if the variate is less than a defined threshold [7, Clause 5.15.1.1]. We assume that the threshold is one; arbitrary thresholds are part of future work.

Due to the mapping scheme, as we discussed in [1], when $N_{SLD}^{TX} > 1$, two UEs that pick resource indices that produce identical values for $b_j^{(i)}$ in a given period, i, will transmit in the same subframes for all N_{SLD}^{TX} transmissions during that period. Thus we can model the discovery resource pool as shown in Fig. 1, where each element of the grid is associated with a unique value of n_{PSDCH}.

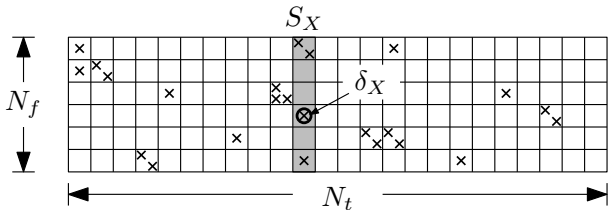


Fig. 1. The discovery resource pool model, showing transmissions from UEs in the D2D group, $\mathcal{G}$, and indicating the location of the discovery message δ_X from a UE of interest, UE X, and the set of subframes used by UE X, S_X (Fig. 3 from [1]).

Let $\mathcal{G}$ denote a group of D2D-capable UEs; the number of UEs in $\mathcal{G}$ is N_u, which we assume is constant. We assume that the area occupied by $\mathcal{G}$ is small enough that every UE in $\mathcal{G}$ is able to receive transmissions from every other UE, and that if a UE picks a resource that no other UE picks, its message will be received by all other UEs. In practice, channel effects will introduce a message loss probability, which we are incorporating into the next generation of this model. We also assume that when two or more UEs pick the same resource, the mutual interference will prevent any collided message's being received by other UEs. In practice, some collided messages may be received by some UEs if the Signal to Interference Ratio (SIR) at the receiver is high enough. There has been some work on modeling the effect of SIR on the discovery process, notably the work by Kang and Kang [8] and Bagheri et al. [9]. However, Kang and Kang compute the average number of devices discovered in a given number of periods, while we obtain the CDF for the number of periods to discover all devices in a group, and Bagheri et al. do not consider the half-duplex effect in their analysis.

III. Analytical Model

We characterize the time for a randomly chosen UE, which we denote as UE X, to discover all other UEs in its group. We use a discrete-time Markov chain whose time index t indicates the number of PSDCH periods that have elapsed since the starting time, and whose single state variable, $N_D[t]$, is the number of UEs that have been discovered by UE X at the end of the tth period. Also, we denote the number of undiscovered UEs at the end of the tth period as $N_U[t] = (N_u - 1) - N_D[t]$. The range of possible values for $N_D[t]$ is $0 \le N_D[t] \le N_u - 1$; the starting state is $N_D[0] = 0$ (because UE X has not yet discovered any of the other $(N_u - 1)$ UEs in the group) and the Markov chain's eventual ending state is $\lim_{n\to\infty} N_D[t] = N_u - 1$, which is the Markov chain's sole absorbing state.

A. The Markov Chain Model

We define the state probability vector for $N_D[t]$ to be $\boldsymbol{\pi}[t] = \left[\pi_0[t], \ldots, \pi_{N_u}[t]\right]$, where $\pi_i[t] = \mathsf{Pr}\{N_D[t] = i\}$ for $i = 0, 1, 2, \ldots, N_u - 1$. Since we start with $N_D[0] = 0$ discovered UEs, $\boldsymbol{\pi}[0] = [1, 0, 0, \ldots, 0]$, and $\lim_{n\to\infty} \boldsymbol{\pi}[t] = [0, 0, \ldots, 0, 1]$.

1) The State Transition Matrix: We define the state transition matrix to be $\mathbf{T} = [T_{i,j}]$, where the probability of transitioning from State i to State j is $T_{i,j} = \mathsf{Pr}\{N_D[t] = j \,|\, N_D[t-1] = i\}$, for $i, j \in \{0, 1, 2, \ldots, N_u - 1\}$. Because $N_D[t]$, the number of UEs discovered by UE X, never decreases, $T_{i,j} = 0$ when $i > j$. For $i \le j$, a transition from State i to State j occurs when UE X discovers $(j - i)$ UEs during the tth period. Thus

$$\begin{aligned} T_{i,j} &= \mathsf{Pr}\{D_a[t] = j - i \,|\, N_D[t-1] = i\} \\ &= \mathsf{Pr}\{D_a[t] = j - i \,|\, N_U[t-1] = N_u - 1 - i\}, \end{aligned} \quad (2)$$

where $D_a[t]$ is the number of UEs discovered by UE X during the tth period, and $D_a[0] = 0$.

In Eq. (3), we show $\mathbf{T}$ when $N_u = 4$. The value of $N_D[t-1]$ is to the left of the corresponding row and the value of $N_D[t]$ is above the corresponding column. Since $N_D[t] = (N_u - 1)$ UEs

$$\mathbf{T} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \left[\begin{array}{cccc|c} \mathsf{Pr}\{D_a[t]{=}0 \,|\, N_D[t{-}1]{=}0\} & \mathsf{Pr}\{D_a[t]{=}1 \,|\, N_D[t{-}1]{=}0\} & \mathsf{Pr}\{D_a[t]{=}2 \,|\, N_D[t{-}1]{=}0\} & \mathsf{Pr}\{D_a[t]{=}3 \,|\, N_D[t{-}1]{=}0\} \\ 0 & \mathsf{Pr}\{D_a[t]{=}0 \,|\, N_D[t{-}1]{=}1\} & \mathsf{Pr}\{D_a[t]{=}1 \,|\, N_D[t{-}1]{=}1\} & \mathsf{Pr}\{D_a[t]{=}2 \,|\, N_D[t{-}1]{=}1\} \\ 0 & 0 & \mathsf{Pr}\{D_a[t]{=}0 \,|\, N_D[t{-}1]{=}2\} & \mathsf{Pr}\{D_a[t]{=}1 \,|\, N_D[t{-}1]{=}2\} \\ \hline 0 & 0 & 0 & \underbrace{\mathsf{Pr}\{D_a[t]{=}0 \,|\, N_D[t{-}1]{=}3\}}_{=1} \end{array}\right] \end{array} \tag{3}$$

TABLE I
LIST OF SYMBOLS

Symbol	Definition
$\mathsf{Pr}\{A\}$	Probability of event A
$\mathsf{E}\{Z\}$	Expected value of random variable Z
L_{PSDCH}	Number of subframes spanned by the PSDCH
$M_{RB}^{PSDCH\_RP}$	Number of PRBs occupied by the PSDCH
l_i^{PSDCH}	ith subframe in the PSDCH
m_j^{PSDCH}	jth PRB slot in the PSDCH
N_r	Number of resources in discovery pool
N_f	Number of PRB pairs in discovery pool
N_t	Number of subframe sets in discovery pool
n_{PSDCH}	Discovery resource index
P	PSDCH period duration
$\mathcal{G}$	The set of UEs in a given D2D group
N_u	Number of UEs in D2D group $\mathcal{G}$
UE X	Randomly chosen UE of interest from $\mathcal{G}$
δ_X	Discovery message sent by UE X
S_X	Set of subframes occupied by δ_X
$\mathcal{N}(A)$	Number of occurrences of event A
n	Number of collided discovery messages
t	Index indicating the tth PSDCH period
$P_C(n \,\|\, N_r; N_u)$	Probability of n collisions given N_u UEs using a pool with N_r resources
$N_U[t]$	Number of UEs in $\mathcal{G}$ undiscovered by UE X at the end of the tth period
$N_D[t]$	Number of UEs in $\mathcal{G}$ discovered by UE X at the end of the tth period
M_D, M_U	Values of $N_D[t-1]$ and $N_U[t-1]$ respectively
m_d, m_u	Number of UEs out of M_D and M_U with discovery messages not in S_X
$D_a[t]$	Number of UEs in $\mathcal{G}$ discovered by UE X during tth period
ν	Value taken by $D_a[t]$
$N_{\mathcal{G}\to X}$	Number of periods for UE X to discover all UEs in $\mathcal{G}$
$\mathbf{T}$	Markov state transition matrix
$T_{i,j}$	(i,j)th element of $\mathbf{T}$
$\mathbf{Q}$	Sub-matrix of $\mathbf{T}$
$\mathbf{N}$	Fundamental matrix of $\mathbf{T}$
$n_{i,j}$	(i,j)th element of $\mathbf{N}$
$\widehat{F}_{N_{\mathcal{G}\to X}}[n]$	Estimated CDF of $N_{\mathcal{G}\to X}$
$\hat{\sigma}_{N_{\mathcal{G}\to X}}$	Standard deviation of error in $\widehat{F}_{N_{\mathcal{G}\to X}}[n]$

is an absorbing state, i.e., $\mathsf{Pr}\{D_a[t] = 0 \,|\, N_D[t-1] = N_u - 1\} = 1$ as shown in Eq. (3).

2) Mean Time to Absorption: For $t > 0$, $\pi_j[t] = \sum_{i=0}^{N_u} \pi_i[t-1]T_{i,j}$, and $\boldsymbol{\pi}[t] = \boldsymbol{\pi}[t-1]\mathbf{T}$, so that $\boldsymbol{\pi}[t] = \boldsymbol{\pi}[0]\mathbf{T}^n$. We can get the CDF of $N_{\mathcal{G}\to X}$ since the $(0, N_u - 1)$th element of $\mathbf{T}^n$ is

$$\begin{aligned}(\mathbf{T}^n)_{0,N_u-1} &= \mathsf{Pr}\{N_D[t] = N_u - 1 \,|\, N_D[0] = 0\} \\ &= \mathsf{Pr}\{N_{\mathcal{G}\to X} \le n\}. \end{aligned} \tag{4}$$

$N_D[t] = N_u - 1$ is the lone absorbing state for this Markov chain (i.e., $T_{N_u-1,N_u-1} = 1$), and all other states are transient (i.e., $T_{i,i} < 1$ for $i \neq N_u - 1$). Because the absorbing state is reachable from all other states, this is an absorbing Markov chain. Given that we start in State $N_D[0] = 0$, we can determine the distribution of $N_{\mathcal{G}\to X}$.

To get an expression for the mean number of periods to reach the absorbing state, we use the chain's fundamental matrix, which we derive using the approach given by Grinstead and Snell [10, Section 11.2]. We start by partitioning $\mathbf{T}$ as follows:

$$\mathbf{T} = \begin{bmatrix} \mathbf{Q} & \boldsymbol{r} \\ \mathbf{0}_{N_u-1} & 1 \end{bmatrix}, \tag{5}$$

where $\mathbf{Q}$ is a $(N_u - 1) \times (N_u - 1)$ matrix whose (i,j)th element is $T_{i,j}$, $\boldsymbol{r}$ is a length-$(N_u - 1)$ column vector whose ith element is $r_i = \mathsf{Pr}\{D_a[t] = (N_u - 1) - i \,|\, N_D[t-1] = i\}$, and $\mathbf{0}_{N_u-1}$ is a length-$(N_u - 1)$ all-zero row vector. Thus

$$\mathbf{T}^2 = \begin{bmatrix} \mathbf{Q}^2 & (\mathbf{I} + \mathbf{Q})\boldsymbol{r} \\ \mathbf{0}_{N_u-1} & 1 \end{bmatrix},$$

where $\mathbf{I}$ is the $(N_u - 1) \times (N_u - 1)$ identity matrix. In general, by recursion and the matrix form of the geometric series,

$$\mathbf{T}^k = \begin{bmatrix} \mathbf{Q}^k & (\mathbf{I} - \mathbf{Q}^k)(\mathbf{I} - \mathbf{Q})^{-1}\boldsymbol{r} \\ \mathbf{0}_{N_u-1} & 1 \end{bmatrix}, \tag{6}$$

where $\mathbf{Q}^0 = \mathbf{I}$. The fundamental matrix is

$$\mathbf{N} = [n_{i,j}] = \sum_{\ell=0}^{\infty} \mathbf{Q}^\ell = (\mathbf{I} - \mathbf{Q})^{-1}, \tag{7}$$

where $n_{i,j} = \sum_{\ell=0}^{\infty} \mathsf{Pr}\{N_D[\ell] = j \,|\, N_D[0] = i\}$, is the mean number of times the chain visits transient State j given that it started in transient State i. Thus the mean of the total number of transient state visits after starting in state $N_D[0] = 0$ is

$$\mathsf{E}\{N_{\mathcal{G}\to X}\} = \sum_{j=0}^{N_u-2} n_{0,j}, \tag{8}$$

which is also the mean number of PSDCH periods required to reach the absorbing state.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

B. The Probability of Collided Discovery Messages

We define $P_C(n \mid N_r; N_u)$ to be the probability that N_u UEs using a pool of N_r resources experience n collisions, where we assume that a collision occurs if two or more discovery messages occupy a given resource. (For all values of N_r and N_u, $P_C(1 \mid N_r; N_u) = 0$.) Each UE independently picks a resource at random, which is equivalent to sampling with replacement N_u times from the set $\{1, 2, \ldots, N_r\}$. The number of ways for N_u UEs to choose a set of resources is $N_r^{N_u}$. We consider two cases, based on the ratio of the number of resources to the number of UEs.

1) $N_r \geq N_u$: The probability that no collisions occur, $P_C(0 \mid N_r; N_u)$, is the probability that the UEs will choose resources so that only one UE uses each utilized resource in the pool. The number of ways that this can happen is $(N_r)(N_r - 1)(N_r - 2) \cdots (N_r - N_u + 1)$. Thus,

$$P_C(0 \mid N_r; N_u) = \frac{N_r!/(N_r - N_u)!}{N_r^{N_u}} = \frac{N_u!}{N_r^{N_u}} \binom{N_r}{N_u}. \quad (9)$$

For $2 \leq n \leq N_u$, we use the occupancy vector $\mathbf{x}$ for the set of resources. The occupancy vector as defined by Feller [11] is the ordered length-N_r vector $\mathbf{x} = [x_1, x_2, \ldots, x_{N_r}]$, which indicates the allocation of UEs among the set of resources without identifying which UEs have chosen a particular resource. When there are n collisions, $\mathbf{x}$ has the form

$$\mathbf{x} = [\underbrace{0, 0, \ldots, 0}_{N_r-(N_u-n)-s}, \underbrace{1, 1, \ldots, 1}_{N_u-n}, \underbrace{k_1, k_2, \ldots, k_s}_{s}],$$

where s is the number of resources that are occupied by two or more UEs. The set of occupancy numbers associated with collided UEs thus form an occupancy sub-vector $\mathbf{k} = [k_1, k_2, \ldots, k_s]$ that has the following two properties:

$$k_1 + k_2 + \cdots + k_s = n \quad (10a)$$

$$2 \leq k_1 \leq k_2 \leq \cdots \leq k_s \quad (10b)$$

Next, we define the vector $\mathbf{d}(\mathbf{k}) = [d_1, d_2, \ldots, d_{\mathcal{U}(\mathbf{k})}]$, where $\mathcal{U}(\mathbf{k}) \in \{1, 2, \ldots, s\}$ is the number of distinct elements of $\mathbf{k}$, and d_i is number of occurrences of the ith distinct element of $\mathbf{k}$. For example, if the number of collisions is $n = 17$, then one possible occupancy vector is $\mathbf{x} = [0, 0, \ldots, 0, 1, 1, \ldots, 1, 2, 2, 2, 3, 5, 5]$. In this case, $\mathbf{k} = [2, 2, 2, 3, 5, 5]$, whose distinct elements are 2, 3, and 5; thus $\mathcal{U}(\mathbf{k}) = 3$, and $\mathbf{d}(\mathbf{k}) = [3, 1, 2]$.

The number of ways that N_r resources can be arranged into $\mathcal{U}(\mathbf{k}) + 2$ groups, where each resource in a given group has been chosen by the same number of UEs, is

$$\frac{N_r!}{(N_r - (N_u - n) - s)!\,(N_u - n)!\,d_1!\,d_2! \cdots d_{\mathcal{U}(\mathbf{k})}!} \quad (11)$$

and the number of ways that the N_u UEs can be arranged into $(N_u - n) + s$ groups, where each group corresponds to an occupied resource, is

$$\frac{N_u!}{\underbrace{1!\,1! \cdots 1!}_{N_u-n} k_1!\,k_2! \cdots k_s!} = \frac{N_u!}{k_1!\,k_2! \cdots k_s!}. \quad (12)$$

Taking the product of Eq. (11) and Eq. (12), multiplying both numerator and denominator by $n!$, and simplifying, we get the number of ways that the N_u UEs in $\mathcal{G}$ can choose resources so that there are n collisions that produce the length-s occupancy sub-vector $\mathbf{k} = [k_1, k_2, \ldots, k_s]$:

$$\frac{N_r! \binom{N_u}{n} \binom{n}{k_1, k_2, \ldots, k_s}}{(N_r - (N_u - n) - s)!\,d_1!\,d_2! \cdots d_{\mathcal{U}(\mathbf{k})}!}. \quad (13)$$

To get the probability of n collisions, we must divide the number of ways to arrange N_u resource choices such that there are n collisions by $N_r^{N_u}$. We get the numerator by summing over all possible occupancy sub-vectors $\mathbf{k}$ that produce n collisions. The sub-vector length, s, varies from $s = 1$ (in which case $\mathbf{k} = [n]$) to $s = \lfloor n/2 \rfloor$, since the longest possible occupancy sub-vector is $\mathbf{k} = [2, 2, \ldots, 2, 2]$ if n is even or $\mathbf{k} = [2, 2, \ldots, 2, 3]$ if n is odd. For a given occupancy sub-vector length s, we sum over all sub-vectors $[k_1, k_2, \ldots, k_s]$ that satisfy Eqs. (10a) and (10b).

Summing Eq. (13) over all possible occupancy sub-vectors, and dividing the result by $N_r^{N_u}$, we get the probability of n collisions:

$$P_C(n \mid N_r; N_u) = \frac{\binom{N_u}{n}}{N_r^{N_u}} \sum_{s=1}^{\lfloor n/2 \rfloor} \frac{N_r!}{(N_r - (N_u - n) - s)!} \times \sum_{\substack{\sum_{i=1}^{s} k_i = n \\ 2 \leq k_1 \leq k_2 \leq \cdots \leq k_s}} \frac{1}{d_1!\,d_2! \cdots d_{\mathcal{U}(\mathbf{k})}!} \binom{n}{k_1, k_2, \ldots, k_s} \quad (14)$$

for $2 \leq n \leq N_u$, when $N_u \leq N_r$.

To get the number of occupancy sub-vectors $\mathbf{k}$, we note that creating $\mathbf{k}$ is analogous to distributing n objects into s bins by first putting one object into each bin, and then distributing the remaining $(n - s)$ objects among the s bins so that at least one object goes into each bin. The number of ways to do this is the number of ways to partition a set of $n - s$ identical objects into s non-empty subsets, which is the set partition number $\Pi(n - s, s)$. There is no closed form expression for this number; we must use the following recurrence relation from Martin [12, p. 35]:

$$\Pi(i, j) = \Pi(i - 1, j - 1) + \Pi(i - j, j) \quad (15)$$

where $\Pi(i, j)$ is the number of ways to partition i indistinguishable objects into j non-empty, indistinguishable groups, where $\Pi(i, i) = \Pi(i, 1) = 1$ for all i, and $\Pi(i, j) = 0$ for $j > i$ [12, p. 35].

2) $N_r < N_u$: For the case where there are more UEs than resources, the number of collisions cannot be zero, since it is impossible to distribute the UEs in such a way that there is one UE per resource. The minimum number of collisions occurs when the occupancy vector has the form

$$\mathbf{x} = [\underbrace{1, 1, \ldots, 1}_{N_r-1}, N_u - (N_r - 1)],$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

so that $\mathbf{k} = [N_u - N_r + 1]$. Thus $P_C(n \,|\, N_r; N_u)$ is non-zero for $N_u - N_r + 1 \le n \le N_u$; and when $N_u > N_r$, $N_u - N_r + 1 \ge 2$. Secondly, the maximum possible length for $\mathbf{k}$ is constrained by the fact that the number of initial zeros in the occupancy vector, $N_r - (N_u - n) - s$, cannot be negative, i.e.,

$$s \le N_r - N_u + n. \tag{16}$$

Applying Eq. (10b) to Eq. (10a) gives $2s \le n$. Thus, when $N_r \ge N_u$, Eq. (16) always holds. However, when $N_r < N_u$, Eq. (16) becomes an additional constraint on the length of $\mathbf{k}$ (e.g., if $n = N_u$, then $s \le N_r$), so that the maximum value of s is $\min(N_r - N_u + n, \lfloor n/2 \rfloor)$. Summing Eq. (13) over all possible occupancy sub-vectors and dividing by $N_r^{N_u}$, we get the following general expression for $P_C(n \,|\, N_r; N_u)$:

$$\begin{aligned} &P_C(n \,|\, N_r; N_u) \\ &\quad = \left(\tbinom{N_u}{n} / N_r^{N_u}\right) \sum_{s=1}^{\min(N_r - N_u + n, \lfloor n/2 \rfloor)} \frac{N_r!}{(N_r - (N_u - n) - s)!} \\ &\qquad\qquad \times \sum_{\substack{\sum_{i=1}^{s} k_i = n \\ 2 \le k_1 \le k_2 \le \cdots \le k_s}} \frac{1}{d_1!\, d_2! \cdots d_{\mathcal{U}(\mathbf{k})}!} \binom{n}{k_1, k_2, \ldots, k_s}, \end{aligned} \tag{17}$$

for $\max(2, N_u - N_r + 1) \le n \le N_u$.

If we compare Eq. (14) and Eq. (17), we see that if $N_r \ge N_u$, then $\min(N_r - N_u + n, \lfloor n/2 \rfloor) = \lfloor n/2 \rfloor$, so that Eq. (17) gives $P_C(n \,|\, N_r; N_u)$ for both cases.

C. The Markov State Transition Probabilities

With $P_C(n \,|\, N_r; N_u)$ in hand, we can get the elements of $\mathbf{T}$. We now derive $\mathsf{Pr}\{D_a[t] = \nu \,|\, N_D[t-1] = M_D\}$, the probability that UE X discovers n UEs in the tth period given UE X has discovered M_D UEs already. First, we condition on the set of events where $m_d \le M_D$ discovered UEs and $m_u \le M_U$ undiscovered UEs do not choose resources in S_X, the set of subframes in which UE X transmits[1], so that their discovery messages can be received by UE X. If $m_u < \nu$, $\mathsf{Pr}\{D_a[t] = \nu \,|\, m_d + m_u \text{ not in } S_X\} = 0$. Also,

$$\begin{aligned} &\mathsf{Pr}\{m_d + m_u \text{ not in } S_X\} \\ &\quad = \mathsf{Pr}\{m_d \text{ not in } S_X\}\, \mathsf{Pr}\{m_u \text{ not in } S_X\} \\ &\quad = \tbinom{M_D}{m_d} \left(1 - \tfrac{1}{N_t}\right)^{m_d} \left(\tfrac{1}{N_t}\right)^{M_D - m_d} \\ &\qquad \times \tbinom{M_U}{m_u} \left(1 - \tfrac{1}{N_t}\right)^{m_u} \left(\tfrac{1}{N_t}\right)^{M_U - m_u}. \end{aligned} \tag{18}$$

Applying both of these conditions, we get

$$\begin{aligned} &\mathsf{Pr}\{D_a[t] = \nu \,|\, N_D[t-1] = M_D\} \\ &\quad = \sum_{m_d=0}^{M_D} \sum_{m_u=\nu}^{M_U} \mathsf{Pr}\{D_a[t] = \nu \,|\, m_d + m_u \text{ not in } S_X\} \\ &\qquad \times \tbinom{M_D}{m_d} \tbinom{M_U}{m_u} \left(1 - \tfrac{1}{N_t}\right)^{m_d + m_u} \left(\tfrac{1}{N_t}\right)^{M_D + M_U - m_d - m_u}. \end{aligned} \tag{19}$$

[1]Note that $M_U = (N_u - 1) - M_D$.

We evaluate $\mathsf{Pr}\{D_a[t] = \nu \,|\, m_d + m_u \text{ not in } S_X\}$ in Eq. (19) by conditioning on the value of ρ, the number of resources occupied by the m_d discovered UEs that do not choose resources in S_X, where $0 \le \rho \le m_d$ (we denote the event "m_d discovered UEs occupy ρ resources" as $\{m_d \Rightarrow \rho\}$). First, we prove the following lemma.

Lemma 1: Given an event A that depends on mutually independent events B_1 and B_2, if there exists a set of events $\{C_i\}_{i=1}^{N}$ that are mutually independent ($C_i \cap C_j = \emptyset$ if $i \ne j$) such that for $i = 1, 2, \ldots, N$, $C_i \subseteq B_1$ and $C_i \cap B_2 = \emptyset$, then $\mathsf{Pr}\{A \,|\, B_1 \cap B_2\} = \sum_{i=1}^{N} \mathsf{Pr}\{A \,|\, B_2 \cap C_i\}\, \mathsf{Pr}\{C_i \,|\, B_1\}$.

Proof:

$$\begin{aligned} &\mathsf{Pr}\{A \,|\, B_1 \cap B_2\} \\ &\quad = \frac{\mathsf{Pr}\{A \cap B_1 \cap B_2\}}{\mathsf{Pr}\{B_1 \cap B_2\}} = \sum_{i=1}^{N} \frac{\mathsf{Pr}\{A \cap B_1 \cap B_2 \,|\, C_i\}\, \mathsf{Pr}\{C_i\}}{\mathsf{Pr}\{B_1\}\, \mathsf{Pr}\{B_2\}} \\ &\quad = \sum_{i=1}^{N} \frac{\mathsf{Pr}\{A \cap B_1 \cap B_2 \cap C_i\}}{\mathsf{Pr}\{B_1 \cap B_2 \cap C_i\}} \frac{\mathsf{Pr}\{B_1 \cap C_i\}\, \mathsf{Pr}\{B_2\}}{\mathsf{Pr}\{B_1\}\, \mathsf{Pr}\{B_2\}} \\ &\quad = \sum_{i=1}^{N} \mathsf{Pr}\{A \,|\, B_1 \cap B_2 \cap C_i\}\, \mathsf{Pr}\{C_i \,|\, B_1\}. \end{aligned}$$

Since $B_1 \cap C_i = C_i$ for $i = 1, 2, \ldots, N$, $\mathsf{Pr}\{A \,|\, B_1 \cap B_2\} = \sum_{i=1}^{N} \mathsf{Pr}\{A \,|\, B_2 \cap C_i\}\, \mathsf{Pr}\{C_i \,|\, B_1\}$. ∎

There are $\widetilde{N}_r = N_r - N_f$ resources available to the $m_d + m_u$ UEs that did not choose resources in S_X. Using Lemma 1, we get

$$\begin{aligned} &\mathsf{Pr}\{D_a[t] = \nu \,|\, m_d + m_u \text{ not in } S_X\} \\ &\quad = \sum_{\rho=0}^{m_d} \mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{m_u \text{ not in } S_X\}\big\} \\ &\qquad \times \mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\}, \end{aligned} \tag{20}$$

where $\mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\}$ is the probability that the occupancy vector for $\widetilde{N}_r$ resources has the form

$$[\underbrace{0, 0, \ldots, 0}_{\widetilde{N}_r - \rho}, \underbrace{k_1, k_2, \ldots, k_\rho}_{d_1, d_2, \ldots, d_{\mathcal{U}(\mathbf{k})}}],$$

and where $\mathcal{U}(\mathbf{k})$ is the number of distinct elements of $\mathbf{k} = [k_1, k_2, \ldots, k_\rho]$, d_i is the number of times the ith distinct element of $\mathbf{k}$ appears in $\mathbf{k}$, and $\mathbf{k}$ is subject to the following restrictions:

$$k_1 + k_2 + \cdots + k_\rho = m_d \tag{21a}$$

$$1 \le k_1 \le k_2 \le \cdots \le k_\rho \tag{21b}$$

The number of ways to divide $\widetilde{N}_r$ resources into $\mathcal{U}(\mathbf{k}) + 1$ groups of sizes $\widetilde{N}_r - \rho, d_1, d_2, \ldots, d_{\mathcal{U}(\mathbf{k})}$ is

$$\binom{\widetilde{N}_r}{\widetilde{N}_r - \rho, d_1, d_2, \ldots, d_{\mathcal{U}(\mathbf{k})}} = \frac{\widetilde{N}_r!}{(\widetilde{N}_r - \rho)!\, d_1!\, d_2! \cdots d_{\mathcal{U}(\mathbf{k})}!},$$

and the number of ways to divide m_d discovered UEs into groups of size $k_1, k_2, \ldots, k_\rho$ is

$$\binom{m_d}{k_1, k_2, \ldots, k_\rho} = \frac{m_d!}{k_1!\, k_2! \cdots k_\rho!}.$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The total number of ways to get this particular value of $\mathbf{k}$ is the product of the two multinomials. By tallying the number of ways that all occupancy vectors with ρ non-zero elements occur, and then dividing by $\widetilde{N}_r^{m_d}$, the total number of possible outcomes, we get the probability that m_d discovered UEs that have chosen resources outside S_X occupy ρ resources:

$$\begin{aligned}&\mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\}\\ &\quad= \frac{\rho!}{\widetilde{N}_r^{m_d}}\binom{\widetilde{N}_r}{\rho}\sum_{\substack{\sum_{i=1}^{\rho} k_i = m_d\\ 1\le k_1\le k_2\le\cdots\le k_\rho}} \frac{\binom{m_d}{k_1,k_2,\ldots,k_\rho}}{d_1!\,d_2!\cdots d_{\mathcal{U}(\mathbf{k})}!}. \end{aligned} \tag{22}$$

There are three special cases that apply to Eq. (22):

- If $m_d = 0$, then $\mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\} = 1$ for $\rho = 0$ and $\mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\} = 0$ for $\rho > 0$.
- If $m_d = 1$, then $\mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\} = 1$ for $\rho = 1$ and $\mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\} = 0$ for $\rho \neq 1$.
- If $m_d > 0$, then $\mathsf{Pr}\{m_d \Rightarrow \rho \,|\, m_d \text{ not in } S_X\} = 0$ for $\rho = 0$.

The next step in the derivation of $\mathbf{T}$ is to develop an expression for $\mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{m_u \text{ not in } S_X\}\big\}$ in Eq. (20). UE X will detect one of the m_u undiscovered UEs that did not choose resources in S_X if the undiscovered UE does not choose the same resource as any other UEs, either discovered or undiscovered. To compute $\mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{m_u \text{ not in } S_X\}\big\}$, we first condition on the number of undiscovered UEs out of m_u that use the ρ resources being used by the m_d discovered UEs that chose resources that are not in S_X. We define $N_{O_{u,d}}$ to be the number of undiscovered UEs not in S_X that overlap with discovered UEs. Since $\{N_{O_{u,d}} = \ell\} \subseteq \{m_u \text{ not in } S_X\}$, by Lemma 1,

$$\begin{aligned}&\mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{m_u \text{ not in } S_X\}\big\}\\ &\quad= \sum_{\ell=0}^{m_u} \mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{N_{O_{u,d}} = \ell\}\big\}\\ &\qquad\times \mathsf{Pr}\big\{N_{O_{u,d}} = \ell \,|\, \{m_d \Rightarrow \rho\}\big\}. \end{aligned} \tag{23}$$

We first compute $\mathsf{Pr}\big\{N_{O_{u,d}} = \ell \,|\, \{m_d \Rightarrow \rho\}\big\}$. The event $\{N_{O_{u,d}} = \ell\}$ occurs when ℓ out of m_u undiscovered UEs that are not in S_X choose resources that are among the ρ out of N_r resources that are occupied by the m_d discovered UEs that did not choose resources in S_X. The probability that one of the m_u undiscovered UEs picks one of the ρ resources, given that it is not in S_X, is $(\rho/N_r)/(\widetilde{N}_r/N_r) = \rho/\widetilde{N}_r$. Thus, the probability that ℓ of the m_u UEs picked one of the ρ resources is

$$\begin{aligned}&\mathsf{Pr}\big\{N_{O_{u,d}} = \ell \,|\, \{m_d \Rightarrow \rho\}\big\}\\ &\quad= \binom{m_u}{\ell}\left(\frac{\rho}{\widetilde{N}_r}\right)^{\ell}\left(1 - \frac{\rho}{\widetilde{N}_r}\right)^{m_u-\ell}. \end{aligned} \tag{24}$$

The last component of the state transition probability that remains to be derived is $\mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{N_{O_{u,d}} = \ell\}\big\}$ in Eq. (23). Since ℓ undiscovered UEs out of m_u chose the same resources as some of the m_d discovered UEs, there are $m_u - \ell$ undiscovered UEs that UE X could discover. If UE X discovers ν UEs out of the $m_u - \ell$ undiscovered UEs, where $0 \le n \le m_u - \ell$, then there were $m_u - \ell - \nu$ collisions among the $m_u - \ell$ undiscovered UEs as they chose from the set of $\widetilde{N}_r - \rho$ resources. The probability of this number of collisions is $P_C(m_u - \ell - n \,|\, \widetilde{N}_r - \rho; m_u - \ell)$. Thus we have

$$\begin{aligned}&\mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{N_{O_{u,d}} = \ell\}\big\}\\ &\quad= P_C(m_u - \ell - \nu \,|\, \widetilde{N}_r - \rho; m_u - \ell). \end{aligned} \tag{25}$$

Note that for the collision probability to be non-zero, $0 \le \ell \le m_u - \nu$.

Using Eq. (24) and Eq. (25), and applying the limits on the values of ℓ, we can write Eq. (23) as

$$\begin{aligned}&\mathsf{Pr}\big\{D_a[t] = \nu \,|\, \{m_d \Rightarrow \rho\} \cap \{m_u \text{ not in } S_X\}\big\}\\ &\quad= \sum_{\ell=0}^{m_u-\nu} \tbinom{m_u}{\ell}\left(\tfrac{\rho}{\widetilde{N}_r}\right)^{\ell}\left(1 - \tfrac{\rho}{\widetilde{N}_r}\right)^{m_u-\ell}\\ &\qquad\times P_C(m_u - \ell - \nu \,|\, \widetilde{N}_r - \rho; m_u - \ell). \end{aligned} \tag{26}$$

Inserting Eq. (22) and Eq. (26) into Eq. (20) gives

$$\begin{aligned}&\mathsf{Pr}\{D_a[t] = \nu \,|\, m_d + m_u \text{ not in } S_X\}\\ &\quad= \sum_{\rho=0}^{m_d}\sum_{\ell=0}^{m_u-\nu} \tbinom{m_u}{\ell}\left(\tfrac{\rho}{\widetilde{N}_r}\right)^{\ell}\left(1 - \tfrac{\rho}{\widetilde{N}_r}\right)^{m_u-\ell}\\ &\qquad\times P_C(m_u - \ell - \nu \,|\, \widetilde{N}_r - \rho; m_u - \ell)\\ &\qquad\times \frac{\rho!}{\widetilde{N}_r^{m_d}}\binom{\widetilde{N}_r}{\rho}\sum_{\substack{\sum_{i=1}^{\rho} k_i = m_d\\ 1\le k_1\le k_2\le\cdots\le k_\rho}} \frac{\binom{m_d}{k_1,k_2,\ldots,k_\rho}}{d_1!\,d_2!\cdots d_{\mathcal{U}(\mathbf{k})}!}. \end{aligned} \tag{28}$$

Finally, applying Eq. (28) to Eq. (19), and rearranging the order of summation, we get Eq. (28). Using Eq. (28) in Eq. (2), we can generate the Markov chain's state transition matrix $\mathbf{T}$ by letting $\nu = j - i$ and $M_D = N_u - 1 - i$ for the set of ordered pairs $\{(i,j) \,|\, j = i, i+1, \ldots, N_u - 1;\ i = 0, 1, \ldots, N_u - 1\}$, using the indexing scheme shown in Eq. (3). We can get $\mathsf{E}\{N_{\mathcal{G}\to X}\}$ by applying Eq. (5), Eq. (7), and Eq. (8), and we can get the CDF of $N_{\mathcal{G}\to X}$ by using Eq. (6) and taking the $(0, N_u - 1)$th element of $\mathbf{T}^k$.[2]

IV. Numerical Results

In this section, we validate the theoretical results from Section III, using both Monte Carlo simulations in Matlab and simulation of a group of UEs in NS3. We also use the theoretical model to determine the maximum group size the allows a single UE to discover all other members of the group within a given amount of time, with a given level of certainty.

For the Monte Carlo simulations, we used a resource pool composed of $N_t = 5$ subframe sets and $N_f = 10$ PRB pairs. The Monte Carlo simulation consisted of $N_{\text{runs}} = 50$ runs,

[2]We note that the computational cost associated with these results can be significant. A desktop using an Intel Xeon CPU with a 3.3 GHz clock rate and 16 GBytes of Random Access Memory (RAM) produced Fig. 2 in about 3000 s, with the Monte Carlo results taking up approximately 2 s.

$$\mathsf{Pr}\{D_a[t] = \nu \,|\, N_D[t-1] = M_D\}$$

$$= \sum_{m_d=0}^{M_D} \sum_{m_u=\nu}^{M_U} \binom{M_D}{m_d} \binom{M_U}{m_u} \left(1 - \frac{1}{N_t}\right)^{m_d+m_u} \left(\frac{1}{N_t}\right)^{M_D+M_U-m_d-m_u} \sum_{\rho=0}^{m_d} \frac{\rho!}{\widetilde{N}_r^{m_d}} \binom{\widetilde{N}_r}{\rho} \left[\sum_{\substack{\sum_{i=1}^{\rho} k_i = m_d \\ 1 \le k_1 \le k_2 \le \cdots \le k_\rho}} \frac{\binom{m_d}{k_1, k_2, \ldots, k_\rho}}{d_1!\, d_2! \cdots d_{\mathcal{U}(\mathbf{k})}!} \right]$$

$$\times \sum_{\ell=0}^{m_u-\nu} \binom{m_u}{\ell} \left(\frac{\rho}{\widetilde{N}_r}\right)^{\ell} \left(1 - \frac{\rho}{\widetilde{N}_r}\right)^{m_u-\ell} P_C(m_u - \ell - n \,|\, \widetilde{N}_r - \rho; m_u - \ell). \tag{28}$$

with $N_{\text{trials}} = 100$ trials per run. The UE group contained $N_u = 51$ UEs, including UE X. Each trial consisted of a sequence of periods in which the number of undiscovered UEs was initialized to $N_u - 1$ and the number of discovered UEs was set to zero. In each period, a $N_f \times N_t$ matrix was populated with randomly placed messages associated with the discovered and undiscovered UEs that were represented using complex numbers, each of whose real part was the number of undiscovered UEs and whose complex part was the number of discovered UEs. One column was chosen at random to contain δ_X; all messages in this column were lost by setting all column elements to zero. The simulation determined the number of successful discoveries of new UEs by counting the matrix elements whose real part was equal to unity. The simulation then adjusted the numbers of discovered and undiscovered UEs and moved to the next period, using the reduction of the number of undiscovered UEs to zero as the stopping criterion.

We plot the theoretical and simulation-based CDFs together in Fig. 2. Because $N_{\mathcal{G}\to X}$ is a discrete random variable, the CDF assumes the staircase form seen in the figure, with point discontinuities indicated by pairs of closed circles and open circles that are connected by dashed lines. Fig. 2 shows uncertainty in the simulation results using two methods. We compute the estimated CDF at a given index value n as

$$\widehat{F}_{N_{\mathcal{G}\to X}}[n] = \frac{1}{N_{\text{runs}}} \sum_{i=1}^{N_{\text{runs}}} \widehat{F}_{N_{\mathcal{G}\to X},i}[n] = \frac{1}{N_{\text{runs}}} \sum_{i=1}^{N_{\text{runs}}} \frac{\mathcal{N}(\{N_{\mathcal{G}\to X_{i,j}}\}_{j=1}^{N_{\text{trials}}} \le n)}{N_{\text{trials}}}, \tag{29}$$

where $\widehat{F}_{N_{\mathcal{G}\to X},i}[n]$ is the estimate of $F_{N_{\mathcal{G}\to X}}[n]$ based on the trials that compose the ith run, $\{N_{\mathcal{G}\to X_{i,j}}\}_{j=1}^{N_{\text{trials}}}$ is the set of simulation outputs generated during the ith run, and where $\mathcal{N}\{\{N_{\mathcal{G}\to X_{i,j}}\}_{j=1}^{N_{\text{trials}}} \le n\}$ is the number of simulation trial outputs during the ith run that were less than or equal to n. From the set of simulation results, we can create a set of pointwise 95 % confidence intervals which appear as light gray bars in the figure, and whose limits for a given value of n are $\widehat{F}_{N_{\mathcal{G}\to X}}[n] \pm 1.96\, \hat{\sigma}_{N_{\mathcal{G}\to X}} / \sqrt{N_{\text{runs}}}$, where $\hat{\sigma}^2_{N_{\mathcal{G}\to X}}$ is the variance of the simulation results and is

$$\hat{\sigma}^2_{N_{\mathcal{G}\to X}} = \frac{1}{N_{\text{runs}} - 1} \sum_{i=1}^{N_{\text{runs}}} (\widehat{F}_{N_{\mathcal{G}\to X},i}[n] - \widehat{F}_{N_{\mathcal{G}\to X}}[n])^2.$$

Fig. 2 also shows a confidence envelope based on Massart's refinement of the Dvoretsky-Kiefer-Wolfowitz (DKW) inequality, which states that, given a set of random variates $\{X_i\}_{i=1}^{N}$ that are drawn from a distribution with CDF F_X and that produce an empirical CDF $\widehat{F}_X$, the probability that the true and empirical CDFs are separated by more than $\varepsilon \in \mathbb{R}^+$ over their entire support has the following upper bound [13]:

$$\mathsf{Pr}\{\sup_{x\in\mathbb{R}} |\widehat{F}_X(x) - F_X(x)| > \varepsilon\} \le 2e^{-2N\varepsilon^2}. \tag{30}$$

Note that the upper bound on the excursion probability in Eq. (30) decreases to zero for any $\varepsilon > 0$ as N increases, and we can easily extend the inequality to discrete random variables, since the resulting CDF is still defined over the whole real line.

The 95 % confidence interval associated with the DKW inequality is actually a confidence envelope over the entire domain of interest, whose upper and lower bounds are defined by the empirical CDF plus or minus the error offset ε. Since we are interested in interval bounds that result in the theoretical curve lying entirely within the confidence envelope with 95 % probability, we can set the upper bound in Eq. (30) equal to 0.05, the probability that the theoretical CDF deviates from the empirical CDF by more than ε, and then solve for ε, giving

$$\varepsilon = \sqrt{\log(2/0.05)/(N_{\text{runs}} N_{\text{trials}})}, \tag{31}$$

since the empirical CDF is constructed from $N = N_{\text{runs}} N_{\text{trials}}$ variates. We plot the resulting confidence envelope $\widehat{F}_{N_{\mathcal{G}\to X}}[n] \pm \varepsilon$ in Fig. 2 using dark gray bars. Note that the pointwise 95 % confidence intervals are contained within the 95 % confidence envelope, which is a looser interval because it covers the entire domain. Note also that the pointwise confidence intervals are widest near the median and become narrower in the distribution's tails.

The resulting plots in Fig. 2 show excellent agreement between the theoretical and simulation results, with the theoretical curve lying entirely within the narrower pointwise confidence intervals over the entire domain. We also used Eq. (8) to get the associated theoretical expected value of $N_{\mathcal{G}\to X}$, which is $\mathsf{E}\{N_{\mathcal{G}\to X}\} = 11.5566$ PSDCH periods. The 95 % confidence interval for the corresponding estimated expected value of $N_{\mathcal{G}\to X}$ that we obtained from the Monte Carlo simulations is $\mathsf{E}\{N_{\mathcal{G}\to X}\} = 11.5988 \pm 0.1007$ PSDCH periods, which also indicates close agreement.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

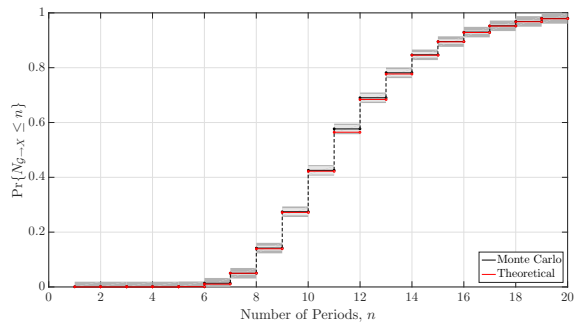


Fig. 2. CDF plots using Eq. (28) and corresponding Monte Carlo simulation results, with pointwise and envelope 95 % confidence intervals shown, plotted versus n, the number of PSDCH periods, for $N_t = 5$ subframe sets, $N_f = 10$ PRB pairs, and $N_u = 51$ UEs.

For the validation using NS3, we examined six scenarios. We examined two resource pool configurations: $N_f = 5$ PRB pairs and $N_f = 10$ PRB pairs, with $N_t = 10$ subframe sets in both cases. For each resource pool configuration, we considered three D2D group sizes: 10 UEs, 20 UEs, and 30 UEs. For each scenario, we performed 10 runs, with 500 trials per run. In each trial, we generated a set of uniformly distributed UEs within an area sufficiently small to that each UE could receive messages from every other UE (e.g., a 10 000 m^2 (100 m $\times$ 100 m) square). Each UE transmitted a discovery message in every period using a randomly chosen pool resource, and we recorded the number of periods required for the UE to discover all of the other UEs in the group. We used these results to generate a sample CDF for each run, and we averaged the ensemble over all runs to produce our estimate of the CDF of $N_{\mathcal{G}\to X}$.

We show the results for the six cases in Fig. 3, with 95 % confidence envelopes for each case, and the corresponding theoretical CDF plotted in each subfigure for comparison. These NS3 results also agree closely with the theoretical model. The results also illustrate the effect of increasing the size of the discovery resource pool. Comparing Figs. 3a-3c with Figs. 3d-3f shows that doubling N_r produces a noticeable leftward shift in the CDF for each group size, and that the CDFs for the various group sizes are closely spaced when $N_r = 100$ resources, while the CDF associated with $N_u = 30$ UEs indicates degraded performance due to increased collisions by UEs using the smaller pool.

Using the CDF, we can calculate the maximum group size that achieves a desired level of performance. Using the Monte Carlo simulations that we used to produce Fig. 2, we plot in Fig. 4 the 50 %, 90 %, and 99 % quantiles of $N_{\mathcal{G}\to X}$ versus N_u for $N_r = 1000$ resources and $N_t = 20$ subframe sets, and show 95 % confidence intervals in the figure. Since $N_{\mathcal{G}\to X}$ is a discrete random variable, a given design constraint will produce a range of values for N_u, as shown in the figure. However, if we require that a randomly chosen UE discover all other UEs in the group within a certain number of PSDCH periods with a given probability, then we would use the largest value of N_u that satisfies this condition. For example, requiring all UEs to be discovered within 7 PSDCH periods with a probability of 0.9 allows a maximum group size between 200 UEs and 300 UEs, using Fig. 4.

To determine the maximum group size with greater precision, we would generate a plot like Fig. 5, which shows the 90 % quantile produced by Monte Carlo simulation, with 95 % confidence intervals shown. Because of the uncertainty in these results, the estimated maximum group size is $(322-340)$ UEs, since 322 UEs is the largest value of N_u whose confidence interval is restricted to 7 PSDCH periods, and 340 UEs is the largest value of N_u whose confidence interval includes 7 PSDCH periods. A conservative design would use the lower end of the range as the upper bound.

V. Summary and Future Work

In this paper, we developed a Markov chain model of the discovery process at a single UE and obtained closed form expressions for the state transition probabilities. Using these in the fundamental matrix allows us to produce the distribution of the number of PSDCH periods required for a UE to discover all of the other UEs in its group, assuming all devices are half-duplex. We validated our results using two approaches: Monte Carlo simulations in Matlab and simulations of groups of UEs in NS3. We showed how to use the model to obtain the quantiles of the discovery time as a function of the group size and pool parameters, which allows one to determine the maximum number of UEs that can use a given resource pool while ensuring that the probability that all UEs are discovered within a given time is below a desired threshold. We also showed that the half-duplex effect means that performance improvements come from adding subframe sets to pools rather than PRBs; future full-duplex UEs will allow performance improvements by expanding pools in either domain.

As we noted previously, this model assumes that messages are lost only during collisions, and that collisions always produce losses. A future version of the model will incorporate the effect of path loss, fading, and shadowing, and will allow for partial recovery of collided messages in high-SIR cases.

References

[1] D. Griffith and F. Lyons, "Optimizing the UE transmission probability for D2D direct discovery," in *2016 IEEE Global Telecommunications Conference (GLOBECOM 2016)*, December 2016.

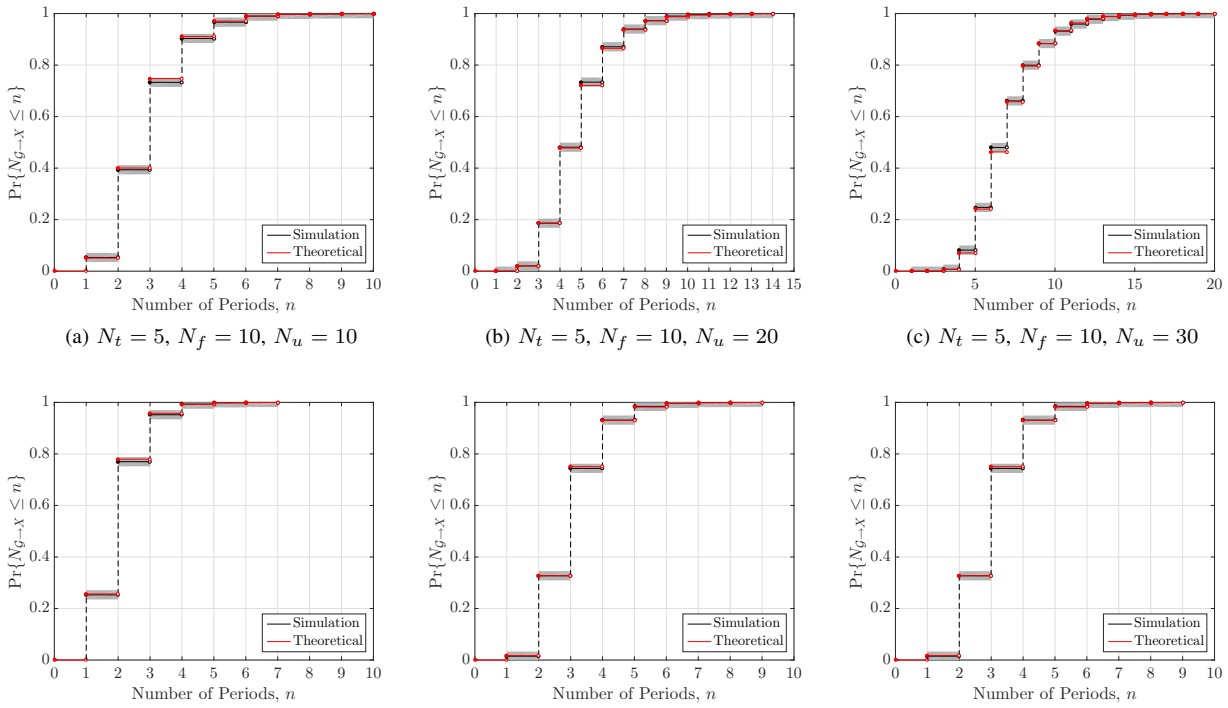
[2] M. G. Sarret, G. Berardinelli, N. H. Mahmood, B. Soret, and P. Mogensen, "Can full duplex reduce the discovery time in D2D communication?" in *2016 International Symposium on Wireless Communication Systems (ISWCS)*, Sept 2016, pp. 27–31.

[3] Z. Lin, L. Du, Z. Gao, L. Huang, X. Du, and M. Guizani, "Analysis of discovery and access procedure for D2D communication in 5G cellular network," in *2016 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2016, pp. 1–6.

[4] Q. Zhang and D. Liu, "On the hopping pattern design for D2D discovery," in *IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, Sept 2014, pp. 1–6.

[5] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 3rd Generation Partnership Project (3GPP), TS 36.213 V12.7.0, September 2015. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.213/36213-c70.zip

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2



(a) $N_t = 5$, $N_f = 10$, $N_u = 10$ (b) $N_t = 5$, $N_f = 10$, $N_u = 20$ (c) $N_t = 5$, $N_f = 10$, $N_u = 30$

(d) $N_t = 10$, $N_f = 10$, $N_u = 10$ (e) $N_t = 10$, $N_f = 10$, $N_u = 20$ (f) $N_t = 10$, $N_f = 10$, $N_u = 30$

Fig. 3. CDF plots using Eq. (28) and corresponding NS3 simulation results, with envelope 95 % confidence intervals shown, plotted versus n, the number of PSDCH periods.

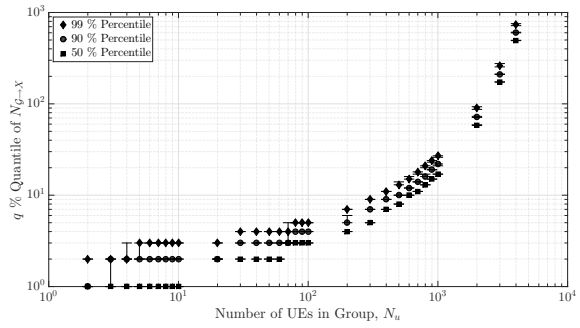


Fig. 4. Plot of the 50 %, 90 %, and 99 % quantiles of $N_{\mathcal{G}\to X}$ generated from Monte Carlo simulations, plotted versus UE group size, N_u. In both plots, $N_r = 1000$ resources and $N_t = 20$ subframe sets. Simulation results include 95 % confidence intervals.

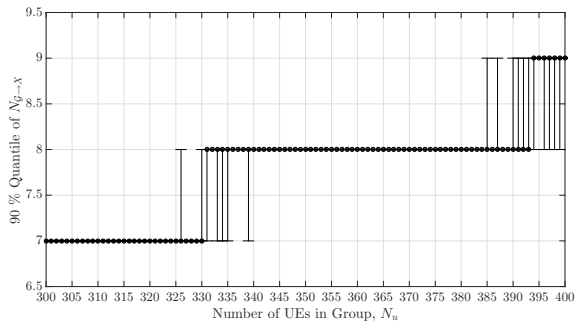


Fig. 5. Plot of the 90 %, quantile of $N_{\mathcal{G}\to X}$ generated from Monte Carlo simulations, plotted versus UE group size, N_u, where $N_r = 1000$ resources and $N_t = 20$ subframe sets. Simulation results include 95 % confidence intervals.

[6] ——, "Study on LTE Device to Device Proximity Services; Radio Aspects," 3rd Generation Partnership Project (3GPP), TR 36.843 V12.0.1, March 2014. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.843/36843-c01.zip

[7] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.321 V12.7.0, September 2015. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.321/36321-c70.zip

[8] H. Kang and C. Kang, "Performance analysis of device-to-device discovery with stochastic geometry in non-homogeneous environment," in *2014 Int. Conf. Information and Communication Technology Convergence (ICTC)*, October 2014, pp. 407–412.

[9] H. Bagheri, P. Sartori, V. Desai, B. Classon, M. Al-Shalash, and A. Soong, "Device-to-device proximity discovery for LTE systems," in *2015 IEEE Int. Conf. Communication Workshop (ICCW)*, June 2015, pp. 591–595.

[10] C. M. Grinstead and J. L. Snell, *An Introduction to Probability: Second Revised Edition*. Providence, RI: American Mathematical Society, 1997.

[11] W. Feller, *An Introduction to Probability Theory and Its Applications, Volume I*. New York: John Wiley & Sons, Inc., 1968.

[12] G. E. Martin, *Counting: the Art of Enumerative Combinatorics*. New York: Springer-Verlag, 2001.

[13] P. Massart, "The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality," *The Annals of Probability*, vol. 18, no. 3, pp. 1269–1283, July 1990. [Online]. Available: http://dx.doi.org/10.1214/aop/1176990746

AP Selection Algorithm with Adaptive CCAT for Dense Wireless Networks

Yena Kim[1], Mun-Suk Kim[1], SuKyoung Lee[2], David Griffith[1], and Nada Golmie[1]

[1]Wireless Networks Division, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA
[2]Department of Computer Science, Yonsei University, Seoul, Korea

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

***Abstract*—Wireless Local Area Networks (WLANs)-enabled devices are now everywhere and their rapid spread has created dense deployment environments. For such dense WLANs, the High Efficiency WLAN Study Group (HEW SG) was formed, and as an extension of their activity, effort on standardization of IEEE 802.11ax Task Group (TG) was initiated. The goal of the TG on IEEE 802.11ax is to improve per-station (STA) throughput of dense WLANs in the presence of interfering sources. To attain this aim, the TG is currently working on Clear Channel Assessment Threshold (CCAT) adjustment. As the CCAT is increased, more concurrent transmissions are permitted, leading to more interference. By using a small CCAT, the amount of interference can be reduced, but the transmission opportunity decays. Thus, we present an algorithm that adjusts CCAT based on the co-channel interference and transmission opportunity for network capacity improvement in dense WLANs. In addition, traffic load may not be fairly shared by all serving Access Points (APs) due to the typical Received Signal Strength (RSS)-based AP selection algorithm. In this paper, therefore, we propose an AP selection algorithm that chooses both AP and CCAT providing the highest achievable throughput for a STA by considering the co-channel interference and the traffic load status in dense WLANs. Simulation results show that our proposed algorithm achieves better performance in terms of the average per-STA throughput and Jain's Fairness Index (JFI) in dense wireless networks with various scenarios.**

I. Introduction

Wireless Local Area Networks (WLANs) [1] are pervasively implemented to provide users with broadband wireless connectivity. Due to its ease of deployment, convenience, and cost efficiency, WLANs are becoming more and more dense. The proliferation of WiFi equipped devices will continue to drive growth in deployed WLANs. However, dense deployment of WLANs causes significantly increased overall interference, and as a result a significantly lowered achievable throughput. Thus, it is sensible to consider technologies that can resolve or mitigate deteriorated throughput of dense WLANs.

In this context, the High Efficiency WLAN Study Group (HEW SG) was formed in May 2013, and as an extension of their activity, IEEE 802.11ax Task Group (TG) has started in May 2014 [2]. This group is targeting ways to enhance IEEE 802.11 physical (PHY) and Medium Access Control (MAC) layers in 2.4 GHz and 5 GHz band with a focus on improving spectrum efficiency and achieving a 4-fold throughput increase compared with IEEE 802.11ac-2013 in high density scenarios. To attain this aim, the 802.11ax TG is currently working on Clear Channel Assessment Threshold (CCAT) adjustment as one of the main issues under consideration [3].

Since WLAN transmission is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), each station (STA) examines the status of the channel prior to transmission by comparing the measured received energy in the wireless channel with the CCAT. The STA attempts channel access only if the measured energy level of the channel is less than the CCAT indicating that the channel is idle; otherwise, the STA backs off and waits for a random period of time. In IEEE 802.11ax, CCAT is adjusted based on the Received Signal Strength (RSS) level of the beacon signal from the associated AP, R, as $(R-m)$ dBm every predefined time period, where m represents a margin [3].

As with this CCAT algorithm in dense WLANs, a STA located near to its respective AP has a higher CCAT, i.e., lower Carrier Sensing (CS) range. It increases the transmission opportunity, but more concurrent transmissions are permitted, leading to more interference. When using a small CCAT, the amount of interference can be reduced but the transmission opportunity decays. It is noteworthy that in the dense deployment of WLANs, the multiple WLANs are operated on the same channel due to limited non-overlapped channels, and thus the network throughput is mainly limited by co-channel interference. Therefore, we propose to adjust CCAT based on both the amount of co-channel interference and transmission opportunity for network capacity improvement in high density WLAN environments.

In addition, traffic load in dense networks may not be fairly shared by all serving APs due to the uncoordinated nature of AP selections among STAs. More specifically, STAs typically select and associate with an AP with the highest RSS. Thus, in this paper, we propose an AP selection algorithm that chooses a combination of AP and CCAT providing the highest achievable throughput for a STA by considering the co-channel interference and the traffic load status in high density WLAN environments.

The rest of this paper is organized as follows. Section II describes our motivations and summarizes related works. Section III describes our AP selection algorithm. In Section IV, we report the results of simulation experiments that demonstrate the performance improvements of the proposed algorithm in terms of per-STA throughput and Jain's Fairness Index (JFI) [4]. Finally, we conclude the paper in Section V.

Golmie, Nada; Griffith, David; Kim, Munsuk; Kim, Ye Na; Lee, SuKyoung.
"AP Selection Algorithm with Adaptive CCAT for Dense Wireless Networks."
Paper presented at 2017 IEEE Wireless Communications and Networking Conference (WCNC),
San Francisco, CA. March 19, 2017 - March 22, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

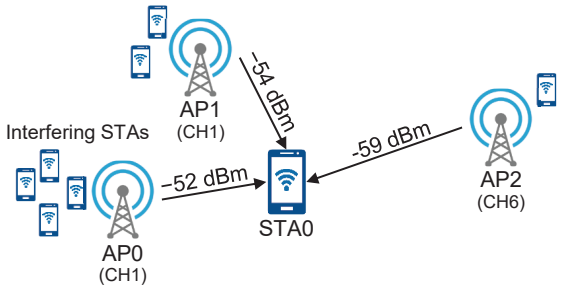


Fig. 1: Example of RSS-based AP selection.

II. Related Work and Motivation

The impact of CCAT on the performance of the wireless network has been studied by a number of researchers [5]–[14]. The authors in [5]–[7] attempt to identify the optimum CCAT in wireless networks and conclude that the fixed CCAT should be optimized to improve the network throughput performance.

Thus, several algorithms have been proposed for optimal CCAT [8]–[14]. Yang *et al.* show that MAC overhead has a significant impact on the choice of the optimal CCAT and then choose the optimal CCAT based on the MAC overhead [8]. The authors in [9] and [10] propose a heuristic algorithm that tunes CCAT according to the varying network conditions, i.e., packet loss. Park *et al.* also use packet error rate to update CCAT, but not differentiate between various cause of packet loss [11]. The authors in [12] and [13] propose a centralized algorithm for adjusting CCAT based on loss differentiation. In their work, all the STAs use the same CCAT. Haghani *et al.* assume that an AP periodically broadcasts its Busy/Idle (BI) signal to the STAs [14]. In their work, individual STA uses the BI signal from the AP in order to adjust their CCAT. Despite the fact that these CCAT tuning algorithms [8]–[14] shows that their algorithms enhance the aggregate throughput in wireless networks, they consider no dense environments.

For dense WLAN deployments, IEEE 802.11 TG suggests a CCAT adaptation algorithm to optimize spatial reuse and enhance throughput performance [2] [3]. Their algorithm dynamically changes the CCAT based on the RSS of the beacon signal of the associated AP. In [15]–[18], the demonstration of CCAT adaptation algorithm is carried out, and it is found that significant per-STA throughput enhancement can be achieved. The throughput are evaluated in dense scenario [19] by adjusting threshold values. However, these algorithms adjust CCAT only for certain AP although other available APs within STAs' vicinity exist. Therefore, we select AP and CCAT jointly. In our approach, as tuning CCAT and selecting an optimal AP, we aim to maximize per-STA throughput. Furthermore, we aim to increase fairness among WLANs.

The current technique of AP selection is for a STA to choose the AP with the strongest RSS. This simple RSS-based method causes each STA to suffer from degraded throughput due to channel contention when a large number of users are crowded together. Furthermore, this method fails when a large number of STAs and APs are deployed densely due to the co-channel interference among the APs and the STAs.

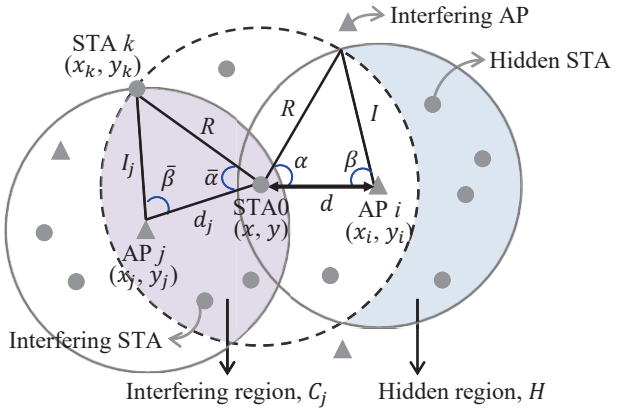


Fig. 2: System model.

Fig. 1 shows an example of RSS-based AP selection. STA0 sees three available APs (denoted as AP0, AP1, and AP2, respectively) in Fig. 1, where the RSS values from these APs observed at STA0 are -52 dBm, -54 dBm, and -59 dBm, respectively. In the RSS-based algorithm, STA0 chooses AP0 based on the RSS value. However, we see that the number of interfering APs and STAs, which all operate on the same channel, are one AP and six STAs for channel 1 and one STA for channel 6, respectively. Thus, selecting AP2 could yield a higher throughput because of less number of interfering APs and STAs (i.e., less interference). If the interference levels are almost the same for three APs, the throughput depends on the APs' load. To reflect these multiple factors into estimating the throughput, we utilize the channel utilization at APs, the RSS from the APs, and the locations of STAs and APs.

III. AP Selection in Dense WLAN Environments

In this section, we describe our proposed AP selection algorithm for dense WLANs.

A. System Model

We consider a network topology consisting of multiple STAs and APs, as shown in Fig. 2. In Fig. 2, the circles with dash line and solid line represent the CS range of STA0 and the coverage areas of the APs, respectively. All STAs are uniformly deployed over the 2-D plane with the density of λ.

In Fig. 2, d represents the distance between STA0 and AP i. Let r_i denote the measured RSS of AP i observed at STA0. We can obtain d using the log-distance power law model [20] as

$$d \cong 10^{(P_0 - r_i)/10\gamma}, \tag{1}$$

where P_0 is the signal strength at the distance l_0 from the AP and γ is the path loss exponent. Typically, l_0 is set to 1 m [20].

I_j and R in Fig. 2 represent the radii for AP j's coverage and STA0's CS range, respectively. Using the same approach as for obtaining d in Eq. (1), we calculate I_j and R as $I_j = 10^{(P_0-\theta)/10\gamma}$ and $R = 10^{(P_0-P_c)/10\gamma}$, where θ is the minimum RSS value required for transmitting a packet from STA0 to AP i and P_c is the CCAT of STA0.

Assume that STA0 associates with AP i and AP j uses the same channel as AP i in Fig. 2. An aggressive CS allows the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

existence of hidden STAs, i.e., the CS range of STA0 does not cover the area of AP i completely. The area where the hidden STAs are located is called hidden region, H. The STAs in H are capable of starting new transmissions since they are out of the CS range. The simultaneous transmission in H can disturb the reception of AP i. The transmissions of the STAs connected to AP j which are located in the CS range of STA0 also interfere the transmissions of STA0. The area where the interfering STAs are located is called interfering region, C_j.

Based on d, I_j, and R, we first estimate H and C_j. Using H and C_j, we then derive the average number of hidden STAs, n_h, and the mean channel utilization of interfering STAs in AP j, $\bar{u}_j$. We will use n_h and $\bar{u}_j$ when estimating the achievable throughput for the AP selection algorithm in next subsection.

1) Number of Hidden STAs: H is within the coverage area of AP i, but outside the CS range of STA0, and is given by

$$H = I^2\pi - \left\{ \left(\frac{R^2\pi^2}{\alpha} - \frac{R^2 \sin 2\alpha}{2} \right) + \left(\frac{I_j^2\pi^2}{\beta} - \frac{I_j^2 \sin 2\beta}{2} \right) \right\}, \tag{2}$$

where α and β are illustrated in Fig. 2 and are given by $\alpha = \cos^{-1}\left(\frac{R^2+d^2-I_j^2}{2 \cdot R \cdot d}\right)$ and $\beta = \cos^{-1}\left(\frac{I_j^2+d^2-R^2}{2 \cdot I_j \cdot d}\right)$.

By using Eq (2), the average number of hidden STAs in the hidden area, n_h, is given by

$$n_h = \frac{H}{I^2\pi} n_i, \tag{3}$$

where n_i is the number of STAs connected to AP i and we can obtain n_i from the beacon of the AP i.

2) Channel Utilization of Interfering STAs: The interfering STA k is placed at (x_k, y_k) such that $\sqrt{(x_k - x_j)^2 + (y_k - y_j)^2} \leq I_j$ and $\sqrt{(x_k - x)^2 + (y_k - y)^2} \leq R$. We then have C_j as follows

$$C_j = \left(\frac{R^2\pi^2}{\bar{\alpha}} - \frac{R^2 \sin 2\bar{\alpha}}{2} \right) + \left(\frac{I_j^2\pi^2}{\bar{\beta}} - \frac{I_j^2 \sin 2\bar{\beta}}{2} \right) \tag{4}$$

where $\bar{\alpha} = \cos^{-1}\left(\frac{R^2+d_j^2-I_j^2}{2 \cdot R \cdot d_j}\right)$ and $\bar{\beta} = \cos^{-1}\left(\frac{I_j^2+d_j^2-R^2}{2 \cdot I_j \cdot d_j}\right)$ in Fig. 2. By using Eq. (1), d_j can be calculated.

Let $\hat{u}_j$ be the WiFi channel utilization of the interfering AP j. STA0 can obtain $\hat{u}_j$ from the AP j through the beacon frame. Since the STAs are placed randomly according to a uniform distribution, the mean channel utilization of interfering STAs connected to AP j, $\bar{u}_j$ is given by

$$\bar{u}_j = \frac{C_j}{I_j^2\pi} \hat{u}_j. \tag{5}$$

B. Throughput Estimation

In this subsection, we estimate the achievable throughput by using n_h and $\bar{u}_j$, which we derived in the previous subsection. Our goal is to estimate the achievable throughput at each AP available to STA0 for all the APs. Let S denote the throughput per STA, defined as the fraction of time the channel is used to successfully transmit payload bits. Letting $E[L]$ be the average packet payload size, we are now able to express S as the ratio

$$S = \frac{P_{\text{success}} E[L]}{P_{\text{success}} T_s + P_{\text{collision}} T_c + P_{\text{idle}} \sigma}, \tag{6}$$

where T_s and T_c represent the average time the channel is sensed busy because of a successful transmission and a collision, respectively. σ is the duration of an empty slot. $T_s = H_{\text{phy}} + H_{\text{mac}} + T_{\text{frame}} + SIFS + ACK + DIFS$ and $T_c = H_{\text{phy}} + H_{\text{mac}} + T_{\text{frame}} + DIFS$, where T_{frame} is the average time to transmit the frame payload, and H_{phy} and H_{mac} are the average transmission times of PHY and MAC headers, respectively [1].

Let P_{success}, P_{idle}, and $P_{\text{collision}}$ denote the probabilities that a successful transmission occurs in a time slot, all STAs are idle in a time slot, and there is a collision in a time slot, respectively. Hence,

$$\begin{aligned} P_{\text{success}} &= n_c p (1-p)^{n_c - 1} (1 - P_L) \\ P_{\text{idle}} &= (1-p)^{n_c} \\ P_{\text{collision}} &= 1 - P_{\text{success}} - P_{\text{idle}} \end{aligned} \tag{7}$$

where n_c is the average number of STAs in the CS range of STA0.

Let $\bar{A}_i$ be a set of interfering APs that use the same channel with AP i, consisting of AP j. Using Eq. (5), the number of competing STAs, n_c, in Eq. (7) depends on the utilization of STAs associated to interfering AP j and is given as

$$n_c = \lambda R^2 \pi \big[1 - \prod_{j \in \bar{A}_i} (1 - \bar{u}_j) \big]. \tag{8}$$

In Eq. (7), p denotes the probability that a STA transmits in a given time slot and P_L is the Packet Error Rate (PER) caused by collisions with STA0. Let $\hat{u}$ and u be the WiFi channel utilization (i.e., the fraction of time that the channel is active) and the channel utilization of STA0 itself, respectively. STA0 can obtain $\hat{u}$ from the AP i. Thus, we can express p as [21]

$$p = \frac{u\sigma}{(1-\hat{u})T_b + \hat{u}\sigma}, \tag{9}$$

where $T_b = (1 - P_L)T_s + P_L T_c$ is the average time that the WiFi channel is sensed busy.

Let ρ be the average packet generation rate at STA0. The utilization u can be expressed in terms of ρ as

$$u = \frac{1}{\rho}(T_s + \frac{P_L}{1 - P_L} T_c), \tag{10}$$

which is limited by $p = \frac{2(1-2P_L)}{(1-2P_L)(W+1) + P_L W (1-(2P_L)^\eta)}$ in saturated conditions where W is the initial contention window size and η is computed from the maximum window size, $2^\eta W$, since the utilization stabilizes to the saturation value when the system reaches saturation [21] [22].

By substituting Eq. (10) in Eq. (9), we can obtain

$$p = \frac{T_b}{\rho(1 - P_L)\{(1-\hat{u})T_b + \hat{u}\sigma\}}. \tag{11}$$

In the network allowing the existence of hidden STAs, the PER is dominated by the hidden STAs. In this way, we ignore the PER due to simultaneous transmission in the interfering range [23]. Thus, P_L is determined by the average number of STAs in the hidden region. The condition of a successful transmission occurs is that there are no concurrent

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

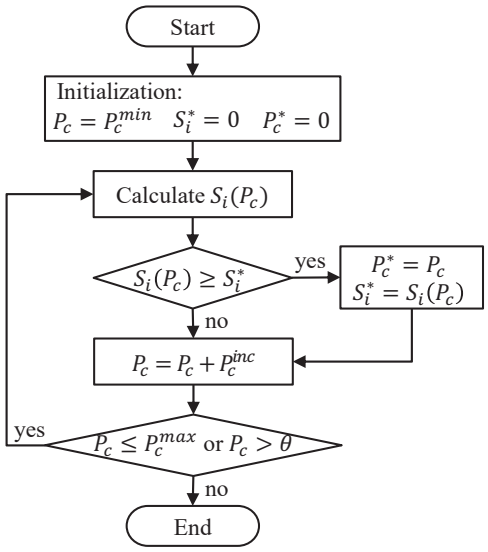


Fig. 3: Flow chart of the proposed algorithm used for each AP

transmissions in the hidden region. From the principle of CSMA, if STA0 starts a transmission, all STAs within the related CS range will not start new transmissions. Therefore, we have $P_L = n_h/n_c$ [23].

C. AP Selection Algorithm

In this subsection, we propose an AP selection algorithm that chooses AP and CCAT as considering the co-channel interference and the channel utilization to increase the achievable throughput and JFI. To do this, we introduce two new parameters for each available AP i within STA0's vicinity: S_i^* and P_c^* to indicate the maximum achievable throughput and the CCAT when STA0 achieves S_i^*, respectively. We also define a function $S_i(P_c)$ which returns the achievable throughput and is affected by the value of P_c ($P_c^{min} \leq P_c \leq P_c^{max}$).

Let A denote a set of the available APs consisting of AP i. When STA0 turns the WiFi interface on, STA0 scans for IEEE 802.11 beacon frames from APs ($\in A$). For each AP i, as shown in Fig. 3, STA0 estimates $S_i(P_c)$ as varying P_c and chooses S_i^* and P_c^* according to the following procedure:

1) Initialize as $P_c = P_c^{min}$, $S_i^* = 0$, and $P_c^* = 0$.
2) Calculate $S_i(P_c)$ by using Eq. (6).
3) Compare $S_i(P_c)$ and S_i^*. If $S_i(P_c) \geq S_i^*$ that means P_c shows higher throughput, set $P_c^* = P_c$ and $S_i^* = S_i(P_c)$; otherwise, go to step 4).
4) Increment P_c by P_c^{inc}. If $P_c \leq P_c^{max}$ or $P_c > \theta$, return to step 2); otherwise, return to step 1) for next AP i ($\in A$).

After the above procedure, STA0 can obtain tuples of <AP i, S_i^*, P_c^* > for APs ($\in A$). Among the tuples, STA0 chooses the tuple that shows the highest achievable throughput, then connects to the AP with the CCAT.

IV. Performance Evaluation

In this section, we evaluate the performance of our algorithm in dense WLAN environments in terms of the average per-STA throughput and Jain's Fairness Index (JFI).

TABLE I: Parameter values for simulation [1] [3]

Parameter	Value	Parameter	Value
MAC & PHY	IEEE 802.11ac	Frequency	5 GHz
Bandwidth	20 MHz	Packet size	1024 bytes
H_{phy}	192 μs	H_{mac}	28 bytes/r
T_{frame}	512 bytes/r	$SIFS$	10 μs
ACK	304 μs	$DIFS$	50 μs
P_c^{min}	-82 dBm	P_c^{max}	-40 dBm

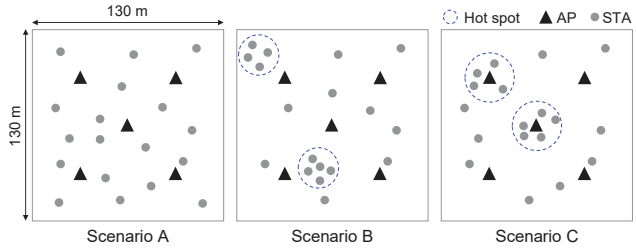


Fig. 4: Simulation scenarios.

A. Simulation Environment

We evaluate the performance of the proposed algorithm through simulations using Network Simulator Version 3 (NS-3) [24]. We compare the average per-STA throughput of the proposed algorithm with that of DSC in which CCAT is set to $R - m$ as mentioned above [3]. We also measure the fairness among STAs with JFI as $\frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2}$, where there are n STAs and x_i is the throughput for the ith connection [4]. JFI is a value between 0 (unfair) and 1 (fair), and JFI is maximum when all STAs receive the same allocation.

Based on the dense networks scenarios defined by IEEE 802.11ax WG [2] [19], we consider a small enterprise of 130 m x 130 m with the bottom left coordinates, (x,y) = (0,0), and the top right coordinates, (x,y) = (130,130), as shown in Fig. 4. In the simulation area, we place five APs at (65,65) m, (40,40) m, (40,90) m, (90,40) m, and (90,90) m, respectively. We vary the number of STAs from 30 to 300.

We take IEEE 802.11ac protocol to evaluate the performance of the proposed algorithm. The average packet generation rate at each STA is randomly chosen from 10 to 20 Mbps. The parameters of PHY and MAC layers used in the simulation are listed in Table I.

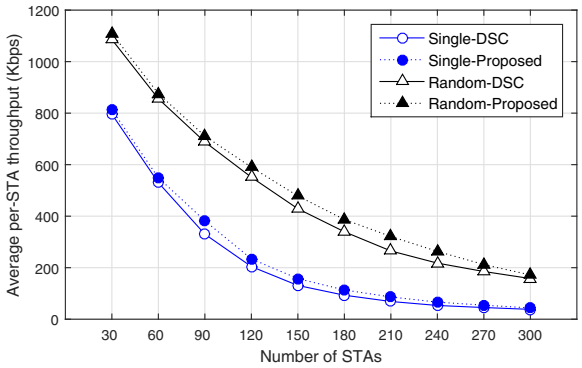
We consider two channel assignment strategies: 1) Single channel: all APs choose the same channel, for example, channel 6 by default; and 2) Random channel: each AP independently selects one of the two non-overlapped channels.

Fig. 4 shows three different scenarios for STA distribution based on uniform distribution. In *Scenario A*, STAs are uniformly distributed within the simulation area (see Fig. 4(a)). In *Scenarios B* and *C*, half of the STAs are uniformly distributed within the scenario, while the other half are uniformly distributed within circular hot spots of 10 m radius. We define hot spot as high density of STAs. The hot spots are located at two random position and at two random APs for *Scenarios B* and *C*, respectively (see Figs. 4(b) and (c)).

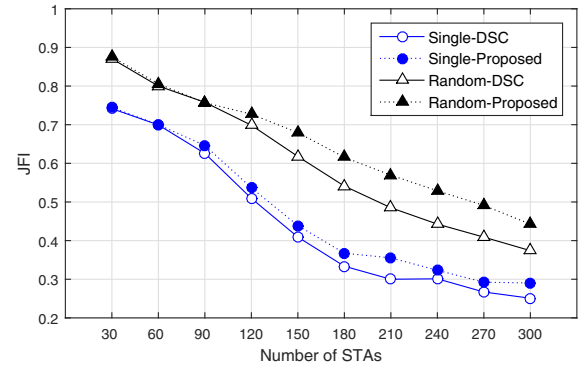
For each simulation scenario, the simulation time is 100 s,

Golmie, Nada; Griffith, David; Kim, Munsuk; Kim, Ye Na; Lee, SuKyoung.
"AP Selection Algorithm with Adaptive CCAT for Dense Wireless Networks."
Paper presented at 2017 IEEE Wireless Communications and Networking Conference (WCNC),
San Francisco, CA. March 19, 2017 - March 22, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

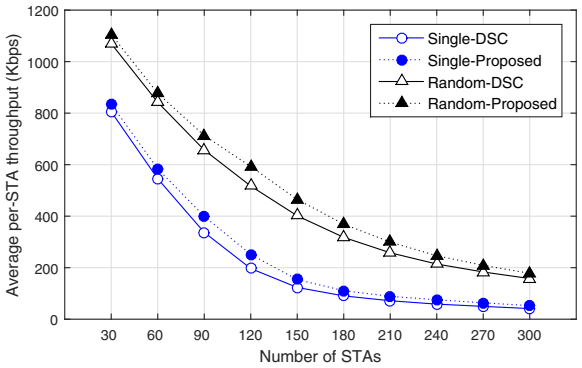


(a) Average per-STA throughput

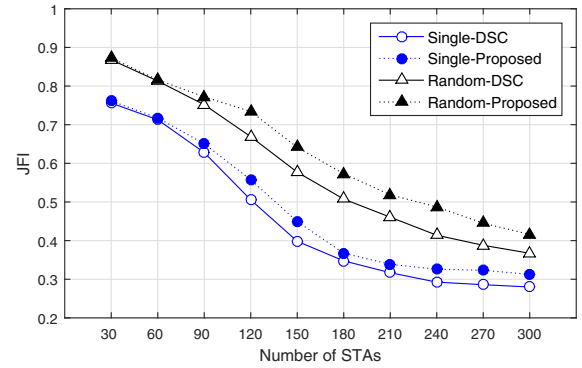


(b) JFI

Fig. 5: Average per-STA throughput and JFI in Scenario A.



(a) Average per-STA throughput



(b) JFI

Fig. 6: Average per-STA throughput and JFI in Scenario B.

and the results are obtained via averaging values from 50 different runs with different seeds.

B. Simulation Results

Fig. 5 plots the variation of the average per-STA throughput and JFI according to the number of STAs when DSC and the proposed algorithm are simulated using single and random channel assignments in Scenario A. It can be seen from Fig. 5(a) that the proposed algorithm achieves better performance than the DSC algorithm in terms of the average per-STA throughput. In addition, we can see from Fig. 5(b) that, as the throughput decreases, JFI gets lower for both algorithms, but the proposed algorithm has higher fairness than DSC regardless of channel assignments and the number of STAs. This is because our algorithm chooses the AP with the optimal CCAT that would provide the highest achievable throughput while DSC selects the AP based only on RSS.

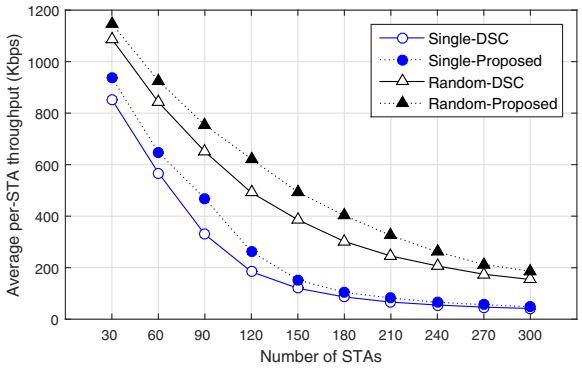
Fig. 6 plots the variation of the average per-STA throughput and JFI according to the number of STAs when DSC and the proposed algorithm are simulated using single and random channel assignments in Scenario B. The throughput and JFI in Scenario B show almost the same behavior as that in Scenario A, indicating that the proposed algorithm performs achieves better performance than DSC. Compared to Scenario A, however, the throughput gain due to the proposed algorithm in Scenario B is larger than that in Scenario A, in where STAs are uniformly distributed. This is because, in DSC, the STAs within hot spots in Scenario B choose a near AP while the STAs in the proposed algorithm chooses the AP considering both the traffic load status and the co-channel interference without considering RSS.

Fig. 7 plots the variation of the average per-STA throughput and JFI according to the number of STAs when DSC and the proposed algorithm are simulated using single and random channel assignments in Scenario C. As with Scenarios A and B, the proposed algorithm in Scenario C outperforms DSC in terms of the throughput and JFI, while the performance gain due to the proposed algorithm is the largest in all scenarios. This is due to the fact that the distances between the AP and the STAs in hot spots is the shortest in Scenario C, meaning essentially that STAs in DSC set CCAT based on the RSS of near AP, resulting in low transmission opportunities at heavily populated APs, while the proposed algorithm considers both the co-channel interference and the traffic load status.

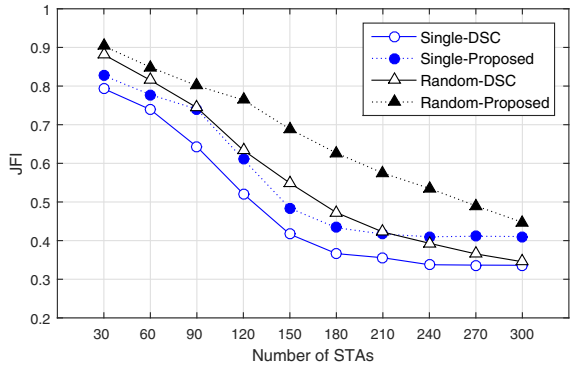
As can be seen in Figs. 5-7, the performance in random channel assignment is better than that in single channel assignment for all scenarios. It is because, as STAs that use the same channel increase, there will be more contention, and hence it results in decreased throughput and JFI.

As shown in Fig. 8, to see the effect of AP locations on the performance of our algorithm and DSC, we measure the average per-STA throughput in Scenario C when APs are randomly distributed. We can see from Figs. 7(a) and 8 that our algorithm achieves better performance than DSC

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2



(a) Average per-STA throughput



(b) JFI

Fig. 7: Average per-STA throughput and JFI in Scenario C.

regardless of the AP locations. We can also observe from the figures, the throughput with random AP locations has slightly higher performance than that with fixed AP locations. It is because the average distance between APs when APs are randomly placed is larger than that when the AP locations are fixed, resulting in decrease of interference.

V. Conclusion

We have presented an AP selection algorithm that chooses a combination of AP and CCAT providing the highest achievable throughput for a STA by considering both the co-channel interference and traffic load status for dense wireless networks. Simulation results showed that the proposed algorithm increases the average per-STA throughput and JFI regardless of the number of STAs and channel assignment strategies in three scenarios with the fixed AP locations. We have also shown that the proposed algorithm outperforms DSC when APs are randomly distributed.

References

[1] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification,* Std., 2012.
[2] IEEE 802.11ax TG. Available: http://www.ieee802.org/11/Reports/tgax_update.htm.
[3] G. Smith (DSP Group), "Dynamic Sensitivity Control V2," doc. IEEE 802.11-13/1012r4, Nov. 2013.
[4] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems," Digital Equipment Corporation, DEC-TR-301, Tech. Rep., 1984.

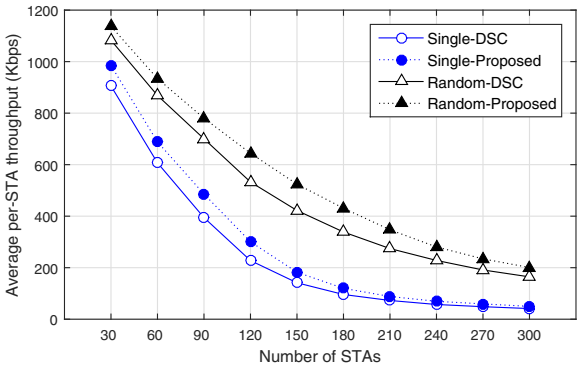


Fig. 8: Average per-STA throughput when APs are randomly distributed in Scenario C.

[5] J. Deng, B. Liang, and P.K. Varshney, "Tuning the Carrier Sensing Range of IEEE 802.11 MAC," in *Proc. IEEE GLOBECOM*, 2004.
[6] J. Zhu, X. Guo, L.L. Yang, and W.S. Conner, "Leveraging Spatial Reuse in 802.11 Mesh Networks with Enhanced Physical Carrier Sensing," in *Proc. IEEE ICC*, 2004.
[7] H. Ma, H.M.K. Alazemi, and S. Roy, "A Stochastic Model for Optimizing Physical Carrier Sensing and Spatial Reuse in Wireless Ad Hoc Networks," in *Proc. IEEE MAHSS*, 2005.
[8] X. Yang and N. Vaidya, "On Physical Carrier Sensing in Wireless Ad Hoc Networks," in *Proc. IEEE INFOCOM*, 2005.
[9] Y Zhu, Q. Zhang, Z. Niu, and J. Zhu, "QoS-aware Adaptive Physical Carrier Sensing for Wireless Networks," in *Proc. IEEE WCNC*, Mar. 2007.
[10] Y. Zhu, Q. Zhang, Z. Niu, and J. Zhu, "On Optimal Physical Carrier Sensing: Theoretical Analysis and Protocol Design," in *Proc. IEEE INFOCOM*, 2007.
[11] K.J. Park, L. Kim, and J.C. Hou, "Adaptive Physical Carrier Sense in Topology-Controlled Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 87-97, Jan. 2010.
[12] H. Ma, R. Vijayakumar, S. Roy, and J. Zhu, "Optimizing 802.11 Wireless Mesh Networks Based on Physical Carrier Sensing," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1550-1563, Oct. 2009.
[13] H. Ma, S. Y. Shin, and S. Roy, "Optimizing Throughput with Carrier Sensing Adaptation for IEEE 802.11 Mesh Networks Based on Loss Differentiation," in *Proc. ICC*, 2007.
[14] E. Haghani, M.N. Krishnan, and A. Zakhor, "Adaptive Carrier-Sensing for Throughput Improvement in IEEE 802.11 Networks," in *Proc. GLOBECOM*, 2010.
[15] I. Jamil, L. Cariou, and J.F. Helard, "Improving the Capacity of Future IEEE 802.11 High Efficiency WLANs," in *Proc. ICT*, 2014.
[16] I. Jamil, L. Cariou, and J.F. Helard, "Efficient MAC Protocols Optimization for Future High Density WLANs," in *Proc. IEEE WCNC*, 2015.
[17] K.S. Shin, I.R. Park, J.H. Hong, D.S. Har, and D.H. Cho, "Per-Node Throughput Enhancement in Wi-Fi DenseNets," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 118-125, Jan. 2015.
[18] M.S. Afaqui, E.G. Villegas, E.L. Aguilera, G. Smith, and D. Camps, "Evaluation of Dynamic Sensitivity Control Algorithm for IEEE 802.11ax," in *Proc. IEEE WCNC*, 2015.
[19] IEEE 802.11-14/0980r5, "TGax Simulation Scenarios," July 2014.
[20] T. Rappaport, Wireless Communications - Principles and Practice, *Prentice-Hall*, 1996.
[21] K. Hong, S. Lee, and K. Lee, "Performance Improvement in ZigBee-based Home Networks with Coexisting WLANs," *Pervasive and Mobile Computing*, vol. 19, pp. 156-166, May 2015.
[22] G. Bianchi, "Performance Analysis of IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, Mar. 2000.
[23] Y. Zhu, Q. Zhang, Z. Niu, and J. Zhu, "On Optimal QoS-aware Physical Carrier Sensing for IEEE 802.11 Based WLANs: Theoretical Analysis and Protocol Design," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1369-1378, Apr. 2008.
[24] "*The Network Simulator, NS-3,*" Available: http://www.nsnam.org/.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Exploiting LTE White Space using Dynamic Spectrum Access Algorithms based on Survival Analysis

Timothy A. Hall, Anirudha Sahoo, Charles Hagwood, Sarah Streett
National Institute of Standards and Technology
Email: {tim.hall, anirudha.sahoo, charles.hagwood, sarah.streett}@nist.gov

***Abstract*—In this study, we design and implement two algorithms for dynamic spectrum access (DSA) that are based on survival analysis. They use a non-parametric estimate of the cumulative hazard function to predict the remaining idle time available for secondary transmission subject to the constraint of a preset probability of successful completion. To show that the algorithms are effective in real-world scenarios even at fine time scales, we evaluate their performance using data collected from an LTE band to model primary user activity. The algorithms are run in different configurations, i.e., they are trained and run on a few combinations of data sets. Our results show that as long as the cumulative hazard functions are fairly similar across datasets, the algorithms can be trained on one day's dataset and run on that of another day's without any significant degradation of performance. The algorithms achieve fairly high white space utilization and have a measured probability of interference which always stays below the preset threshold.**

I. Introduction

Dynamic spectrum access (DSA) seems poised to mitigate the problem of spectrum scarcity. In a typical DSA scenario, a primary user (PU) has priority access to a given band. A secondary user (SU) can transmit during unoccupied (idle) periods opportunistically but must vacate when the PU needs the band again. In order to make efficient use of the spectrum in a DSA environment, an accurate and useful model of spectrum occupancy is needed.

Spectrum occupancy refers to whether or not a particular channel or band is occupied. In this paper, we use the term channel to denote the smallest allocable range of frequencies within a particular communications technology, e.g., 180 kHz for LTE. A band is comprised of multiple channels and represents a single service, e.g., there are 50 channels in a 10 MHz LTE uplink band. We model the occupancy of a given channel as a two-state (binary) random process similar to that used by Spaulding and Hagn [1]:

$$X(t) = \begin{cases} 1 & \text{if } P_R(t) > P_{th} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $P_R(t)$ is the signal power observed at the receiver at time t and P_{th} is a threshold value. $X(t) = 1$ represents the occupied state and $X(t) = 0$ represents the unoccupied state.

A. Previous Work

Various models have been proposed in the literature for spectrum occupancy. A two state Discrete-Time Markov Chain (DTMC) has been used to model spectrum occupancy in [2]. However, stationary DTMC models have been found to be inadequate to represent idle and busy periods. Hence, authors in [2] have proposed a time-inhomogeneous DTMC model. Some authors have also used semi-Markov models for spectrum occupancy [3]. This study assumes a general distribution (rather than exponential) for the idle and busy periods of the spectrum. Further, since there are only two states (ON/OFF), the process is also analyzed as an Alternating Renewal Process [3], [4].

Continuous-Time Markov Chain (CTMC) based models have also been used to represent spectrum idle and busy periods. Since some measurement studies have shown that the ON and OFF periods of spectrum are not exponentially distributed, authors in [5], [6], [7] have used semi-Markov models for the purpose.

Model occupancy of adjacent channels has been modeled as a two-dimensional Markov chain by Gibson and Arnett in [8], [9].

Some studies have shown that busy and idle periods of spectrum exhibit negative correlation, i.e., the idle period following a long busy period is typically short and vice-versa [10]. In this study, the authors have proposed time-correlation models for periodic and non-periodic auto-correlation functions.

There have been few models proposed for predicting spectrum occupancy, which is critical to allocating spectrum to the secondary users. The Partially-Observable Markov Decision Process (POMDP) model has been proposed in [11]. The spectrum sharing scheme proposed in [12] is based on prediction of spectrum occupancy by the primary users in terms of the expected remaining OFF time. A two state semi-Markov model proposed in [3] is used to estimate the distribution parameters of ON/OFF periods. Some methodologies proposed in the literature indirectly predict spectrum occupancy by limiting the duration of transmission of the secondary user (SU) to some constraint. In [13], the transmission duration of an SU is based on the maximum bound on probability of interference to the primary user (PU). Residual idle time of an Alternating Renewal Process is used in [4] to indirectly predict reappearance of the PU. Some researchers have used a Restless Multiarm Bandit formulation for opportunistic channel access [14], [15]. Researchers have also looked at pattern mining of spectrum occupancy data to predict channel

U.S. Government work not protected by U.S. copyright

Hagwood, Robert; Hall, Timothy; Sahoo, Anirudha; Streett, Sarah.
"Exploiting LTE White Space using Dynamic Spectrum Access Algorithms based on Survival Analysis."
Paper presented at 2017 IEEE International Conference on Communications (ICC), Paris, France. May 21, 2017 - May 25, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

availability [16], [17].

B. Motivation for Present Work

The motivation behind the present work is threefold. We want to develop a prediction scheme that is robust, flexible and useful even for very fine time scales. We assume centrally coordinated scheduling for the SUs. The scheduler knows when the primary user is no longer active, and when an SU requests a transmission opportunity, the scheduler grants or denies the SU request. Our scheme is not limited to a centralized scheduling architecture, however. It can be used in a carrier sense multiple access (CSMA) system as well. In such a system, the SUs would sense the channel and use our algorithms to predict residual idle time before transmitting as a form of collision avoidance. Analysis and application of prediction schemes presented in this paper to a CSMA based system is beyond the scope of this study.

Most of the stochastic based schemes in the literature either assume a certain distribution (e.g., exponential) of spectrum occupancy data or require that a distribution be fitted to a set of observed data. This study does not have such a requirement. It uses a non-parametric estimate of the cumulative hazard function from historical data to grant dynamic access to the SUs. Hence, our scheme is much simpler to implement in practice.

Finally, most of the DSA schemes in the literature are run over simulated spectrum occupancy data. We ran our algorithms over real spectrum occupancy data to show that they are suitable for implementation on practical systems. We show the effectiveness of our DSA algorithms over LTE Band 17, which is centered at 709 MHz with a 10 MHz bandwidth in the uplink.

We also envision that our scheme (or some variation thereof) may be used in the Spectrum Sharing architecture proposed in the 3.5 GHz band [18]. In this architecture, there will be three tiers of users in the band. First tier users have the highest priority, but they use the band infrequently. The tier two users, called Priority Access Layer (PAL) users, will likely be LTE carriers and have medium priority. When tier 1 and tier 2 users are not present in the band, it can be used by tier 3 users called General Authorized Access (GAA) users. It is conceivable that a PAL user can *sell* its white spaces (idle times) to users who can make use of transmission opportunities of the order of hundreds of milliseconds as long as the interference to PAL users remains below an agreed threshold. These opportunistic users can implement our scheme to exploit PAL white spaces.

Let us now define the prediction problem upon which our DSA algorithms are based more precisely. We are concerned with how long the channel has been unoccupied by the PU and how much longer the channel will remain unoccupied. Specifically, given that the channel has been unoccupied by the PU for duration t and a request from an SU arrives to transmit for a duration τ, what is the probability that the SU will be able to complete the transmission before the PU appears on the channel? Figure 1 illustrates the relationship between the PU and SU.

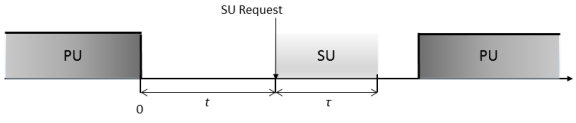


Fig. 1. SU request

The remainder of this paper is structured as follows. Section II formulates the prediction problem in terms of survival analysis, resulting in two algorithms for secondary channel requests. Section III describes the collected data, simulation environment and metrics we used to evaluate the algorithms. Section IV presents our results. Section V interprets the results and discusses future work.

II. Prediction Algorithms

A. Survival Analysis

Survival analysis has been used to analyze statistical properties of the duration of time until an event, such as failure in a mechanical system, occurs [19]. Our prediction problem can be solved by using survival analysis as presented below.

Let $T_1, S_1, T_2, S_2, \ldots$, represent the successive idle and busy periods of the spectrum. Thus T_i and S_i represent the i^{th} idle and busy periods respectively. The T_i's can be thought of as survival times. That is, an idle period survives only until the channel becomes busy again. Let random variable T represent an arbitrary survival time and $0 < p < 1$ an adjustable parameter. Assuming the T_i are independent and identically distributed as T, our prediction problem can be represented by the hypothesis testing problem given by

$$\begin{aligned} \mathcal{H}_0 &: \ P[T \geq t+\tau \mid T \geq t] > p \quad versus \\ \mathcal{H}_1 &: \ P[T \geq t+\tau \mid T \geq t] \leq p \end{aligned} \tag{2}$$

$\mathcal{H}_0$ holds if the idle period, having lasted t units of time, lasts τ more units of time with probability greater than p. Note that p represents the probability of successful transmission for duration τ, given that the channel has been idle for duration t.

The basic functions of survival analysis are the survival function and the hazard function. The survival function at time t is the probability of surviving at least t units of time and is given by

$$S(t) = P[T \geq t] = 1 - F(t) = \int_t^\infty f(s)ds \tag{3}$$

where $f(s)$ and $F(t)$ are the probability density function and cumulative distribution function of T, respectively. The hazard function is the probability of instantaneous failure at time t given survival up to time t and indicates the risk of failure at time t. The hazard function is given by

$$\begin{aligned} h(t) &= \lim_{\delta t \to 0} \frac{P[t \leq T < t+\delta t \mid T \geq t]}{\delta t} \\ &= \lim_{\delta t \to 0} \frac{P[t \leq T < t+\delta t]}{P[T \geq t] \cdot \delta t} \\ &= \frac{1}{P[T \geq t]} \cdot \lim_{\delta t \to 0} \frac{P[t \leq T < t+\delta t]}{\delta t} \\ &= \frac{f(t)}{S(t)} \end{aligned} \tag{4}$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

From (3), it is clear that the derivative of $S(t)$ is $-f(t)$. Hence, (4) can be rewritten as

$$h(t) = -\frac{d}{dt} log S(t) \tag{5}$$

Now integrating both sides of (5) from 0 to t, noting that $S(0) = 1$ and finally taking the exponential on both the sides, we have

$$S(t) = \exp\left(-\int_0^t h(s)ds\right) \tag{6}$$

The function important to us is the cumulative hazard function, defined by $H(t) = \int_0^t h(s)ds, t \geq 0$. Using (6) we have

$$\begin{aligned} P[T \geq t+\tau \mid T \geq t] &= \frac{P[T \geq t+\tau]}{P[T \geq t]} \\ &= \exp\left(-\int_0^{t+\tau} h(s) + \int_0^t h(s)ds\right) \\ &= \exp(-[H(t+\tau) - H(t)]). \end{aligned} \tag{7}$$

Thus, using (7) the hypotheses in (2) can be expressed as

$$\begin{aligned} \mathcal{H}_0 :&\ \ \exp(-[H(t+\tau) - H(t)]) > p \quad versus \\ \mathcal{H}_1 :&\ \ \exp(-[H(t+\tau) - H(t)]) \leq p \end{aligned} \tag{8}$$

Having observed a large sample $T_1, T_2, \ldots, T_n$ of n survival times, a non-parametric estimate of the survival function can be computed using the empirical distribution function, $F_n(t)$ of the data $T_i, i = 1, \ldots, n$, as shown below.

$$S_n(t) = 1 - F_n(t) = 1 - \frac{1}{n}\sum_{i=1}^{n} 1_{T_i < t} \tag{9}$$

where 1_A is the indicator function for event A.

Let $T_{(1)} \leq T_{(2)} \cdots \leq T_{(n)}$ be the ordered $T_i, i = 1, \ldots, n$. Then the survival function at any $T_{(i)}$ can be computed using (9) as follows.

$$\begin{aligned} S_n(T_{(i)}) &= 1 - \frac{1}{n}\sum_{j=1}^{n} 1_{T_j < T_{(i)}} \\ &= 1 - \frac{1}{n} \cdot (i-1) = \frac{n-i+1}{n} \end{aligned} \tag{10}$$

In the above derivation, we used the fact that exactly $(i-1)$ values of T_i are strictly less than $T_{(i)}$. Each $T_{(i)},\ 1 \leq i \leq n$, has an estimated probability of occurence of $\frac{1}{n}$. Hence,

$$f_n(T_{(i)}) = \frac{1}{n} \tag{11}$$

Using (10) and (11) in (4) we have

$$\begin{array}{lcll} h_n(T_{(i)}) & = & \frac{1}{n-i+1} & \text{for } i = 1, 2, \cdots, n \\ h_n(t) & = & 0 & \text{for all other } t \end{array}$$

Using the definition of the cumulative hazard function, an estimate is given by

$$H_n(t) = \sum_{i:T_{(i)} \leq t} \frac{1}{n-i+1} \tag{12}$$

Our test statistic is based on the difference of the cumulative hazard function at two different times. An estimate for the difference of the cumulative hazard function at two different times is given by

$$H_n(t+\tau) - H_n(t) = \sum_{i:t \leq T_{(i)} \leq t+\tau} \frac{1}{n-i+1} \tag{13}$$

Note that this is a form of the well-known Nelson-Aalen estimator for the cumulative hazard function. We used a more general form of $H_n(t)$ to account for duplicate values of $T_{(i)}$, that is, multiple idle times of the same duration [20]. Therefore, after simple manipulation of $\mathcal{H}_0$ in (8), our prediction algorithms are formulated in terms of an approximate test statistic,

$$\text{Reject } \mathcal{H}_0 \text{ if : } \ H_n(t+\tau) - H_n(t) \geq (-\ln p). \tag{14}$$

B. Definition of Algorithms

Below are two formulations of the prediction algorithm. The first is a request to transmit on a channel for duration τ. If the channel is occupied at the time of request, the request is denied. If the channel is not occupied, then the algorithm grants the request if it determines that the probability of a successful transmission (i.e., the probability of completing the transmission without colliding with the PU) is above a given threshold.

Algorithm 1 Request channel for τ seconds

input:
τ - the transmit duration requested
parameters:
$H_n(t)$ - the estimated cumulative hazard function
t_0 - the time elapsed since end of last transmission
p - the probability of successful transmission
output: *Grant* or *Deny*

if *occupied* **then**
  return *Deny*
end if
$\theta := -\ln p$
$W_n := H_n(t_0 + \tau) - H_n(t_0)$
if $W_n < \theta$ **then**
  return *Grant*
else
  return *Deny*
end if

The second algorithm returns the longest estimated duration available for transmission for a request made at a particular time. The time returned is the largest value for which the probability of successful transmission exceeds the given threshold.

Algorithm 2 Request maximum channel availability

parameters:
$H_n(t)$ - the estimated cumulative hazard function
$\{T_{(i)}\}$ - the n ranked idle times used to compute $H_n(t)$
t_0 - the time elapsed since end of last transmission
p - the probability of successful transmission
output: τ - the maximum transmit time available now

if *occupied* **then**
  return 0
end if
$\theta := -\ln p$
Find largest τ in $[0, T_{(n)}]$ such that $H_n(t_0+\tau) - H_n(t_0) < \theta$
return τ

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

III. Evaluation

To evaluate the algorithms, we used real LTE uplink spectrum occupancy data to represent our primary user occupancy. Secondary user requests for spectrum were simulated using a Poisson arrival model, i.e., the SU request inter-arrival times were exponentially distributed.

A. Data Collection

Data was collected in Band 17, a 10 MHz uplink LTE band centered at 709 MHz. A small 10.78 cm rubber duck antenna was connected to an Ettus Universal Sofware Radio Peripheral (USRP) [1] running USRP hardware driver (UHD) version 003.009.001 and GNU Radio version 3.7.9rc1. The output is a 56 point power spectrum computed every 100 ms. Each power spectrum coefficient is an 8 bit signed integer representing a decibel (dB) value rounded to the nearest integer. Each coefficient corresponds to peak power in dB over a 180 kHz range. The middle 50 coefficients correspond to the 50 LTE channels. We applied a noise threshold power value to produce a binary occupancy sequence for each of the 50 channels. We looked at all the different power values collected and picked the $75th$ percentile value as the noise threshold. For the data collected, the threshold turned out to be -67 dB. The idle time distributions for day1 and day2 are presented in Table I.

Data was collected for two continuous 48 hour periods. The first 48 hours ran from 2:00 PM UTC (9:00 AM local time) Monday, February 1, 2016 to 2:00 PM UTC on Wednesday, February 3, 2016. The second dataset covers weekend hours, 2:00 PM UTC Saturday, February 27, 2016 to 2:00 PM UTC Monday, February 29, 2016. Each 48 hour data set is split into two parts, each containing 24 hours of data. Thus, the data captured from Monday 9 AM to Tuesday 9 AM is designated as $day1$ data, and the data captured from Tuesday 9 AM to Wednesday 9 AM is designated as $day2$. Similarly, the weekend data is termed as $wknd1$ and $wknd2$.

B. Simulation

As stated above, an idle period of the spectrum occupancy is a set of one or more consecutive zeros. Each zero represents an idle period with a duration of one sampling interval (100 ms for our experiments). Thus, the T_i values (in terms of sampling interval) are represented as the number of consecutive zeros. Similarly, a busy period is a set of one or more consecutive ones. In our experiments, we have used the occupancy of LTE uplink channel number 5 as our PU traffic. After building the idle and busy periods, we then compute the cumulative hazard function as per (13). We have set the probability of successful transmission (p) to 0.9. Thus, the interference threshold, which is equal to $(1-p)$, is set to 0.1. Note that the PU expects its measured probability of interference (PoI) to be less than this preset interference threshold.

[1]The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

We have evaluated the performance of Algorithm 1 and Algorithm 2 in different configurations as described below. The configurations are denoted as *train_run*, where *train* is the data used for training the algorithm (i.e., the cumulative hazard function is built using this data) and *run* represents the data which is used to run the algorithm. We have four data sets, each of 24 hours duration.

As an example, in configuration $day1\_day1$, the algorithms are trained using $day1$ data, i.e., the cumulative hazard function is built using $day1$ data and then the algorithm is also run on $day1$ data. Results from this configuration validate the effectiveness of survival analysis for opportunistic spectrum access.

When using configuration *day1_day2* the algorithms are trained using day1 data but run on day2 data. This configuration helps us understand how the algorithms perform when the training and running data are from different week days. Note that in practice, the *day1_day1* configuration does not correspond to a realistic scenario, since the training has to happen on some historical data and then the algorithm would run on different data. Hence, this configuration is useful in practice.

Yet another example is configuration *wknd1_day1*. This configuration helps us determine if it is feasible to train the algorithm on a weekend data set and run it on a week day data set.

C. Metrics

We used the following metrics to measure performance of the two algorithms. The first two metrics are common to both the algorithms whereas the remaining four are defined for Algorithm 1 only.

- White Space Utilization (WSU): Given the spectrum occupancy of a channel, White Space Utilization (WSU) of the channel by a secondary user is defined as the fraction of total idle time used by the secondary user for its own transmission. In another words, it is the ratio of total duration of idle time used by the secondary user for its own transmission to the total idle time duration in the spectrum occupancy of the channel.
- PoI: For a given channel, the PoI of the secondary user is defined as the probability that a transmission of the SU collides with that of the PU. Thus, it is the ratio of the number of times an SU transmission collides (or runs into a busy period) with a PU transmission to the total number of SU transmissions over a statistically long observation period.
- Desirable Accept Ratio (DAR): This is defined as the fraction of requests that were accepted and the corresponding transmissions were successful. In these cases the algorithm correctly predicted the remaining idle time.
- Undesirable Accept Ratio (UAR): This is defined as the fraction of requests that were accepted and the corresponding transmissions were not successful, i.e., these transmissions resulted in collision with PU transmission. In these cases the algorithm incorrectly predicted the remaining idle time.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Length (sample inter-val)	Number of occurrences day1 (%)	Number of occurrences day2 (%)	Length (sample interval)	Number of occurrences day1 (%)	Number of occurrences day2 (%)
1	10 331 (39.75 %)	9866 (35.98 %)	(10, 20]	760 (2.92 %)	1093 (3.99 %)
2	8523 (32.78 %)	8784 (32.03 %)	(20, 30]	281 (1.08 %)	447 (1.63 %)
3	2174 (8.36 %)	2475 (9.03 %)	(30, 40]	165 (0.63 %)	258 (0.94 %)
4	957 (3.68 %)	969 (3.53 %)	(40, 50]	124 (0.48 %)	149 (0.54 %)
5	664 (2.55 %)	796 (2.9 %)	(50, 500]	546 (2.1 %)	783 (2.86 %)
6	431 (1.66 %)	491 (1.79 %)	(500, 5000]	64 (0.25 %)	133 (0.48 %)
7	308 (1.18 %)	361 (1.32 %)	(5000, 10 000]	5 (0.02 %)	18 (0.07 %)
8	264 (1.02 %)	302 (1.1 %)	(10 000, 17 871]	3 (0.01 %)	10 (0.04 %)
9	240 (0.92 %)	295 (1.08 %)	(17 871, 132 171]	4 (0.01 %)	0 (0 %)
10	152 (0.58 %)	193 (0.7 %)			

TABLE I
IDLE TIME DISTRIBUTION OF DAY1 AND DAY2 DATA

- Desirable Reject Ratio (DRR): This is defined as the fraction of requests that were rejected and would have resulted in collision with the PU if they were accepted. So, in these cases the algorithm correctly predicted the remaining idle time and rejected the requests.
- Undesirable Reject Ratio (URR): This is defined as the fraction of requests that were rejected and would have resulted in successful transmission if they were accepted. In these cases the algorithm incorrectly predicted the remaining idle time and rejected the requests. This metric represents lost opportunities for the SU.

IV. RESULTS

Figure 2 shows the performance of Algorithm 1 in terms of WSU as average request inter-arrival time varies. As request inter-arrival time increases, WSU decreases since the offered load from the SU decreases. It is interesting to note that the performance of all the configurations in terms of WSU are almost the same. Since the cumulative hazard functions of the different days have almost the same slope for most of the values between $n = 0$ to $n = 100$, the decision to accept or reject a request is almost the same regardless of which day's data is used for training. This leads to nearly the same WSU for different configurations. Thus, Algorithm 1 can be trained using data from any of the days without significantly affecting the WSU of the system.

From Figure 3, we observe that the PoI is always less than the set threshold of 0.1. When we compare the PoI of Algorithm 2 (see Figure 5) we notice that the PoI is an order of magnitude less than that of Algorithm 2. Algorithm 1 only transmits for a fixed duration when the request is granted. In our experiment the fixed duration is 200 ms, which is a relatively short duration. In other words, when the requested duration is short, Algorithm 1 is less aggressive than Algorithm 2. Hence, the probability of an SU transmission colliding with the PU is very low.

Figure 4 shows the performance of Algorithm 2 in terms of WSU. As the average request inter-arrival time increases, the SU exploits less white space for transmission. Hence, the WSU decreases. We also notice that WSUs for the set of configurations which are run on day1 (e.g., day1_day1, day2_day1, wknd1_day1) are higher than those run on day2. This can be explained by studying the cumulative hazard functions of the two days. The cumulative hazard functions of the data set are shown in Figure 6 and Figure 7. Since the range of idle period is very large, we show the $H(\cdot)$ function up to 100 sampling intervals in Figure 6, whereas Figure 7 shows the entire range of idle period. The cumulative hazard functions of day1 and day2 have almost the same slope for idle periods less than 100 sampling intervals. Hence, for a given time of arrival of an SU request, the maximum duration granted would be almost the same for the two days. So, the idle time distribution of the two days has more influence than the $H(\cdot)$ function on the WSU for the two days for grants less than 100 sampling intervals. From Table I, we observe that day1 has some very large idle times. For example, day1 has four idle times in the range $(17\,871, 132\,171]$, which are very large idle times, whereas the maximum idle time of day2 was $17\,871$. This leads to more transmission opportunities for the SU when running over day1 data and gives rise to higher WSU for day1 than for day2. Now for idle periods greater than 100 sampling periods (refer to Figure 7), for day2, the slope of $H(\cdot)$ is very steep, whereas for day1 the slope flattens due to the extremely long run lengths of idle periods. Thus, when the time of arrival of a request falls into an idle period that has lasted longer than 100 sampling intervals, day1 grants longer transmission opportunities. Furthermore, when the elapsed idle time is more than the longest idle period in the training data, the transmission opportunity granted is set to the longest idle period of training data. These two factors also contribute to higher WSU for day1 than day2.

Since the cumulative hazard functions of the four days have almost equal slope for most of the n values less than 100 sampling intervals, training the algorithm on data from any day produces nearly the same WSU for that given day. Thus, the curves for day1_day1, day2_day1 and wknd1_day1 are close to each other.

When we compare the WSU performance of Algorithm 1 with Algorithm 2 for a given request inter-arrival time, we notice that the WSU of Algorithm 1 is much lower than that of Algorithm 2. The fundamental design of the two algorithms gives rise to this behavior. For a given request, Algorithm 2 maximizes the SU transmission duration, whereas Algorithm 1 only checks to see if it can grant a request for a constant transmission duration (200 ms in our experiment). Thus, Algorithm 2 is able to achieve higher WSU.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

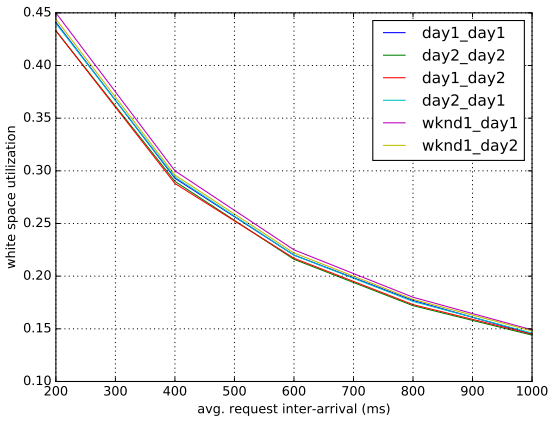


Fig. 2. WSU vs inter-arrival time for Algorithm 1

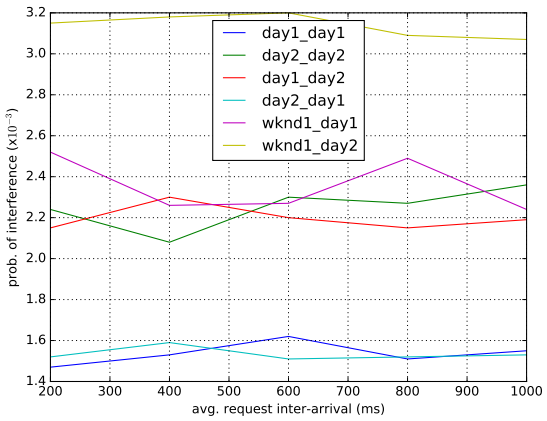


Fig. 3. PoI vs inter-arrival time for Algorithm 1

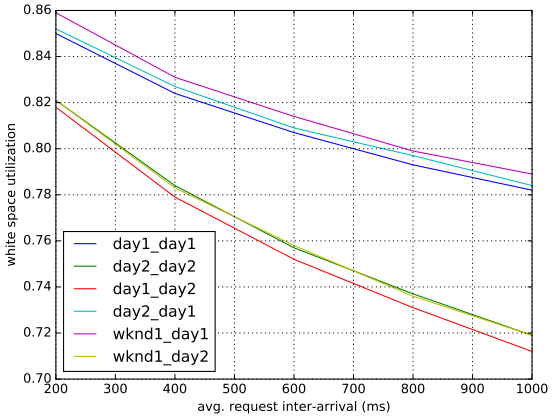


Fig. 4. WSU vs inter-arrival time for Algorithm 2

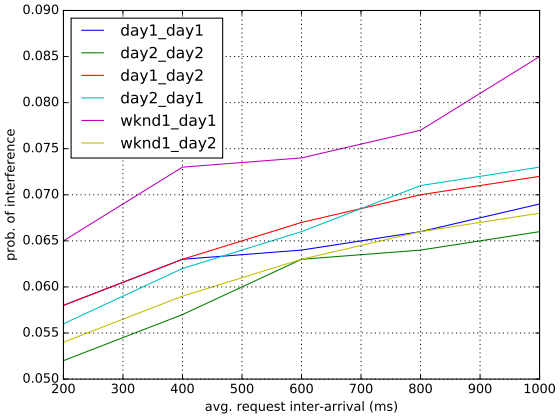


Fig. 5. PoI vs inter-arrival time for Algorithm 2

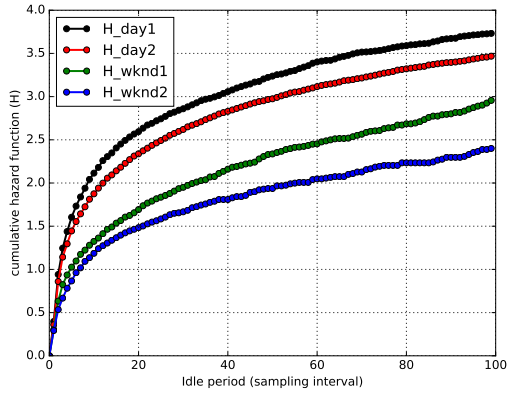


Fig. 6. Cumulative Hazard Function (up to 100 sampling interval)

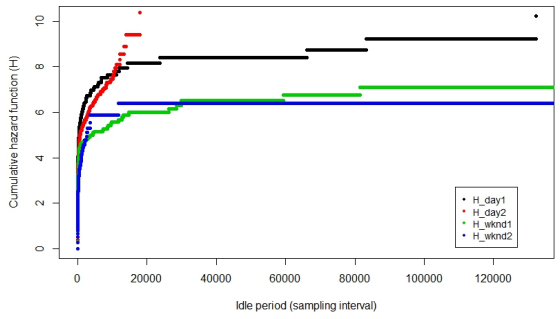


Fig. 7. Cumulative Hazard Function (for the entire range of idle period)

Figure 5 shows the variation of PoI as the inter-arrival time of the requests increases. For all configurations, the PoI is below the set threshold (0.1), thus satisfying the interference constraint of the PU.

We also ran our two algorithms on the weekend data (wknd1 and wknd2 data) sets. We are not able to present the results here due to space limitation, however, the results look very similar to the week day results presented in this paper.

Our results indicate that the algorithms can be trained using any data set and run on another data set as long as the cumulative hazard functions are similar (in terms of slope).

We show the desirable and undesirable accept and reject ratios of Algorithm 1 when the SU request inter-arrival time is 200 ms. The URR is zero for all configurations. Thus

Configuration	DAR (%)	UAR (%)	DRR (%)	URR (%)
day1_day1	78.7	0.2	21.1	0
day2_day1	78.6	0.2	21.1	0
wknd1_day1	81.4	0.4	18.2	0
day2_day2	76.2	0.3	23.5	0
day1_day2	76.2	0.3	23.5	0
wknd1_day2	79.6	0.5	19.9	0

TABLE II
VARIOUS ACCEPT AND REJECT RATIOS FOR REQUEST INTER-ARRIVAL TIME 200 MS FOR ALGORITHM 1

Algorithm 1 has no lost opportunities in all the configurations. The algorithm also has very low UAR, which is good, since this metric shows how well the algorithm avoids making bad decisions in accepting a request. Although we have results for other request inter-arrival times, we are not able to present them due to space limitations. However, those results are equally good.

All our experiment runs were for a very long duration (approximately twenty-four hours). Hence, the number of SU requests were very large. So, the computed performance metrics of the two algorithms (e.g., WSU and PoI) had very little variation across different runs.

V. CONCLUSION AND FUTURE WORK

We introduced DSA algorithms based on survival analysis that make efficient use of white space in an LTE band, even at very fine time scales. They are stochastic but non-parametric and therefore do not require the assumption of a particular distribution. This makes the implementation simple. The tuning parameter for the algorithm is the probability of successfully completing a transmission or, equivalently, the PoI. Thus, it is easy to interpret and directly reflect desired system performance metrics. We used real LTE band occupancy data for the PU activity in our simulations. Our results show that if the cumulative hazard functions are fairly similar (in terms of slope) across different datasets, the algorithms can be trained on one day's dataset and run on another day's dataset without significant degradation of performance. This is a very important property of the algorithms, since in practice, the algorithms will be trained on historical data and then run in real-time. We expect that in actual spectrum sharing systems the PUs will be wary of sharing their spectrum with SUs for fear of too much interference. This is addressed in our algorithms by showing that the PoI is always below the preset threshold in all configurations.

This paper provides an initial performance analysis of the algorithms in an LTE band. Evaluation using datasets collected in different bands at varying locations with other traffic characteristics needs to be done. Other time scales need to be investigated to show the range over which the algorithms are effective. A theoretical performance analysis and comparison with other prediction schemes are needed as well.

Depending on the SU application, alternative forms of the algorithms presented in this paper can easily be developed using the same fundamental approach. One can imagine a form of spectrum requests that includes a maximum or desired transmit time and a minimum acceptable time. The algorithm would then either deny the request or return a grant duration in the requested range. Another form could have the user requesting a minimum initial grant and then the scheduler can add additional follow-on transmission time, if available, once the initial request has elapsed. An adaptive version of the algorithm may be more attractive for implementation on practical systems. It would update the estimated cumulative hazard function as new idle periods appear in the spectrum.

REFERENCES

[1] A. Spaulding and G. Hagn, "On the definition and estimation of spectrum occupancy," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-19, no. 3, pp. 269–280, August 1977.

[2] M. Lopez-Bentez and F. Casadevall, "Discrete-time spectrum occupancy model based on markov chain and duty cycle models," in *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, May 2011, pp. 90–99.

[3] H. Kim, and K. Shin, "Efficient Discovery of Spectrum Opportunities with MAC-layer Sensing in Cognitive Radio Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 533–545, May 2008.

[4] M. Sharma and A. Sahoo, "Stochastic Model Based Opportunistic Channel Access in Dynamic Spectrum Access networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1625–1639, July 2014.

[5] S. Geirhofer, L. Tong, and B. M. Sadler, "Dynamic spectrum access in WLAN channels: Empirical model and its stochastic analysis," in *TAPAS '06 Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum*, August 2006.

[6] ——, "Dynamic spectrum access in the time domain: Modeling and exploiting white space," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 66–72, May 2007.

[7] L. Stabellini, "Quantifying and modeling spectrum opportunities in a real wireless environment," in *2010 IEEE Wireless Communication and Networking Conference*, April 2010, pp. 1–6.

[8] A. Gibson and L. Arnett, "Statistical modelling of spectrum occupancy," *Electronics Letters*, vol. 29, no. 25, pp. 2175–2176, 1993.

[9] ——, "Measurements and statistical modelling of spectrum occupancy," in *HF Radio Systems and Techniques, 1994., Sixth International Conference on*, July 1994, pp. 150–154.

[10] M. L. Lopez-Benitez and F. Casadevall, "Modeling and simulation of time-correlation properties of spectrum use in cognitive radio," in *International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCom)*, June 2011, pp. 1–5.

[11] Q. Zhao, L. Tong, A. Swami and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 589–600, April 2007.

[12] K. W. Sung, S. Kim and J. Zander, "Temporal Spectrum Sharing Based on Primary User Activity Prediction," *IEEE Transactions on Wireless Communications*, vol. 9, no. 12, pp. 3848–3855, December 2010.

[13] A. Plummer, M. Taghizadeh and S. Biswas, "Measurement based bandwidth scavenging in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 19–32, January 2012.

[14] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5588–5611, August 2012.

[15] Y. Gai and B. Krishnamachari, "Decentralized online learning algorithms for opportunistic spectrum access," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, December 2011, pp. 1–6.

[16] S. Yin, D. Chen, Q.Zhang, M. Liu and S. Li, "Mining Spectrum Usage Data: A Large-Scale Spectrum Measurement Study," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1033–1046, June 2012.

[17] P. Huang, C-J. Liu, X. Yang, L. Xiao and J. Chen, "Wireless Spectrum Occupancy Prediction Based on Partial Periodic Pattern Matching," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1925–1934, July 2014.

[18] "Further Notice of Proposed Rule Making," https://apps.fcc.gov/edocs_public/attachmatch/FCC-13-100A1.pdf, April 2014, [Online; accessed 25th March 2015].

[19] R. G. Miller Jr, *Survival analysis*. John Wiley & Sons, 2011, vol. 66.

[20] O. Aalen, O. Borgan, and H. Gjessing, *Survival and event history analysis: a process point of view*. Springer Science & Business Media, 2008.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

A Novel Adaptive Transmission Algorithm for Device-to-Device Direct Discovery

Aziza Ben Mosbah, David Griffith and Richard Rouil
National Institute of Standards and Technology, Gaithersburg, Maryland, USA
{*aziza.benmosbah, david.griffith, richard.rouil*}*@nist.gov*

***Abstract*—In this paper we study and improve one service used for Proximity Services and Device-to-Device (D2D) communications: D2D Direct Discovery. As defined in the Third Generation Partnership Project, for both in-coverage and out-of-coverage cases, resource pool parameters, including the transmission probability (in *UE-Selected* mode), are configured in advance. This means that they are independent of the network conditions and the number of users. Thus, we propose an adaptive algorithm which takes into account the available resources and the number of nearby users as they are being discovered, and adapts the transmission probability accordingly. This algorithm improves the overall performance of the discovery process. It reduces the time needed to complete the discovery within a group of UEs and makes it dynamic and adaptable to changing environments.**

***Index Terms*—Long Term Evolution (LTE), Device-to-Device (D2D), Discovery, Proximity Services (ProSe), Simulations, Performance, Adaptive Algorithm**

I. Introduction

The Third Generation Partnership Project (3GPP) introduced the notion of Proximity Services (ProSe) to Long Term Evolution (LTE) in its Release 12. ProSe enable Device-to-Device (D2D) communications services between nearby User Equipment (UE) [1]. It is incorporated in the existing LTE Advanced services and networks [2] with the goal to offload traffic from the network and provide extra capacity. It also extends the network coverage in scenarios with failed or non-existent infrastructure [3]. One of the new services that was defined to enable this D2D communication was the discovery of nearby users and applications. This service was initially limited to UEs within network coverage for both commercial and public safety usage. In Release 13, it was extended to work in out-of-coverage cases for public safety applications.

In the in-coverage scenarios, the discovery configuration is broadcast to the discovery-eligible UEs through messages from the Evolved Node B (eNB) [4]. However, if no network assistance is used, the UEs use pre-configured parameters. The relevant configuration comprises the discovery period of configurable length (between 0.32 and 10.24 seconds), the resource blocks to use, and the discovery bitmap that indicates which subframes could be used for discovery. It also specifies the number of repetitions (i.e. how often this bitmap is repeated within the discovery period), the number of re-transmissions of the discovery message, and the transmission probability.

In [3], operators are given the option of using either Evolved Packet Core (EPC)-level discovery, so that the core network has more control over the discovery process, or direct discovery. In addition, for privacy reasons, discovery can be either open or restricted. In the latter case, explicit permission is required from the device that is being discovered.

3GPP defined two discovery models where Model A is an unconditional broadcast of announcements described as "Here I am!" sent by UEs, and Model B is based on a request/response process ("Who is there?" "I am here!"). Furthermore, the way discovery resources are allocated defines the type of discovery. Type 1, referred to as "UE-selected", allows the device to autonomously and randomly select the radio resources from the resource pool to transmit the discovery message. In type 2B, noted "scheduled", the eNB provides a dedicated resource allocation for each announcing UE on a UE-specific basis. We are interested in D2D direct discovery, model A, type 1 where UEs do not rely on eNBs for resource selection and vicinity awareness. In such scenarios, independently of the network coverage, the discovery parameters are defined beforehand. They do not take into account the potential diversity in group topologies nor the dynamics introduced by the users' mobility. Therefore, a flexible and adaptive discovery algorithm is needed in order for the discovery process to be executed efficiently and without requiring a large number of resources.

In this paper, we propose an algorithm that allows UEs performing discovery to tune the transmission probability based on the available resources and the number of UEs discovered throughout the discovery process. This allows a quick convergence to the optimal transmission probability, resulting in a faster and more efficient discovery.

This paper is organized as follows: In Section II, we provide a review of the related work and literature. We describe our novel adaptive algorithm and the relevant assumptions we used in Section III. In Section IV, we use system-level simulations to evaluate its performance and efficiency. Finally, we draw the conclusion and outline our future work in Section V.

II. Related Work

The available research on D2D has mostly focused on the communication performance. Nevertheless, there are a few significant contributions in the literature that provide insight into this process. Sun et al. [5] provide a summary on D2D

Ben Mosbah, Aziza; Griffith, David; Rouil, Richard.
"A Novel Adaptive Transmission Algorithm for Device-to-Device Direct Discovery."
Paper presented at IWCMC 2017, Valencia, Spain. June 26, 2017 - June 30, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

synchronization, discovery, and communication, as stated in the 3GPP specifications at the time (March 2014). Xu et al. [6] show that if UEs are using scheduled mode and are within cell coverage, it is easier to avoid collisions and improve discovery. The authors propose to use the eNB to gain knowledge of the UEs that want to participate in D2D, and allocate the resources for discovery based on their position and the number of UEs in the network. Similarly, Choi et al. [7] propose another network-assisted model where the eNB efficiently allocates resources for discovery as it is aware of all the discovery traffic going on in the cell. Also, using the eNBs to improve the discovery process, Ngo et al. [8] use two eNBs to calculate the relative distance between UEs, and presume that this knowledge can be used to accelerate the discovery process. However, these proposals focus only on network-assisted discovery where the eNB is in control, therefore limiting their improvements to in-coverage scenarios.

The most notable contribution to out-of-coverage discovery in the literature is the proposal by Li et al. [9] of a static scheme to control interference. Instead of basing the transmission of discovery messages on a transmission probability, the authors propose using a scheme that replaces the randomness of the probability with predictable deterministic "equivalent" transmissions (i.e., instead of a transmission probability of 0.25 each period, they propose one transmission every 4 periods). However, this model cannot react to changes in the size of the group or in the transmission conditions.

Griffith and Lyons [10] propose a theoretical model that calculates the optimal value of the probability of transmission for a given set of parameters (i.e., number of UEs and resources). The model assumes a prior knowledge of the number of users in the discovery group, which is most likely not the case in reality. In addition, it considers an ideal propagation environment, disregarding fading and interference factors. In this paper, we build upon their research by using the analytical model proposed in [10] as the basis of an adaptive algorithm where the UEs adjust their transmission probabilities over time. By doing this, we manage to first overcome the requirement of knowing the number of UEs to be discovered, and then make the enhanced discovery process work with more realistic propagation environments, taking into account loss and recovery probabilities.

To the best of our knowledge, this work is the first one to propose such an algorithm that improves the overall performance of out-of-coverage discovery by dynamically adjusting one of its parameters to different environments, group topologies, and resource configurations.

III. Adaptive Algorithm

A. System Model

In Table I, we provide a list of symbols we use in this paper. We assume that each UE sends one discovery message (i.e., one announcement) after checking its transmission probability threshold, each time it is supposed to do discovery. For clarity, and without loss of generality, we will consider that the number of applications is equal to the number of UEs. In a general case, some UEs would be interested in monitoring just one or two applications in order to discover their corresponding UEs. However, at the physical layer, the UEs receive all announcements from all surrounding UEs independently of their applications of interest, and the filtering happens in the upper layers. That is why we consider that all UEs are interested in announcing their own application and in monitoring all other applications within the group.

TABLE I: List of Symbols

Symbol	Definition
N_f	Number of resource block pairs available for discovery
N_t	Number of subframes available for discovery
N_r	Total number of resources in discovery pool
UE_X	Randomly chosen UE
n	Number of UEs discovered by UE_X
N_n	Number of new UEs discovered by UE_X
N_o	Number of UEs previously discovered by UE_X
N_u	Number of UEs in the discovery group
N_X	Number of UEs discovered by UE_X plus UE_X itself
θ	Optimal transmission probability for UE_X
UE_n	Late arrival UE

When sending announcements, UEs choose the resources to use from a defined resource pool. The pool is defined by given numbers of subframes N_t and of resource block (RB) pairs N_f. The total number of resources N_r is equal to $N_t \times N_f$.

According to the 3GPP working assumptions [3], the channel used for D2D is half-duplex. So, if a UE transmits a discovery message in one subframe, it cannot receive any other discovery message transmitted by any other UE in that same subframe. Taking this into account, [10] presents an analytical model proving that D2D direct discovery performance can be enhanced using an optimal value of the transmission probability θ, defined by Eq. (1). The use of this value makes the discovery faster.

$$\theta = \frac{2N_r + N_t(N_u - 1) - \sqrt{4N_r(N_r - N_t) + N_t^2(N_u - 1)^2}}{2N_u}. \tag{1}$$

An exception has been identified when the number of UEs is small compared to the number of resources available for discovery: If the condition in Eq. (2) is fulfilled, the optimal value of θ is 1 (meaning that the UEs will transmit discovery announcement messages all the time).

$$N_u < \frac{N_r(N_t - 2) + N_t}{N_t - 1} \quad \text{where} \quad N_t > 1^1. \tag{2}$$

As we can see, the computation of the optimal transmission probability requires prior knowledge of the number of UEs in the group (N_u), which means that in a changing environment the UEs need to learn that information dynamically.

B. Adaptive Discovery Process

We consider a group of N_u users that decide to start using D2D communication at the same time (e.g., when a group of

[1] If $N_t = 1$, UEs would always announce at the same subframe and would never be able to discover each other because of the half-duplex constraint.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

emergency responders arrives at an incident location). They hold discovery-capable equipment, and start sending discovery messages using a pre-configured transmission probability and allocated resources N_f, N_t, and N_r.

We assume that each UE (noted as UE_X) has already detected N_o UEs in previous discovery periods (i.e. $N_o = 0$ at the beginning of the discovery process). At the end of the current period, UE_X successfully receives discovery messages from n different UEs. However, only N_n of those n received discovery messages have never been received before. N_X represents the total number of UEs that UE_X succeeded to discover, including itself. When the discovery process is complete, and if the UE_X succeeded to discover every UE in the group, N_X should be equal to N_u.

$$N_X = N_o + N_n + 1\,; \tag{3}$$

where $N_o < N_u$, $N_n \leq n < N_u$, and $N_X \leq N_u$.

The adjusted transmission probability of UE_X for the next period is the approximation to the nearest non-zero multiple of 0.25 less than or equal to 1 (to conform to the values allowed by 3GPP) of the final result of Eq. (1) and (2) using Eq. (3).

C. Proposed Algorithm

For any given UE_X, the computation of the adjusted transmission probability will follow Algorithm 1.

Data: N_o is the total number of different UEs discovered by UE_X in previous discovery periods
for *any given* UE_X *performing D2D discovery* **do**
  UE_X receives discovery messages from n UEs;
  $N_n = 0$;
  for i *in* $[1, n]$ **do**
    if UE_i *was never discovered before* **then**
      increment N_n;
    end
  end
  $N_X = N_o + N_n + 1$;
  if $N_X > 1$ **then**
    compute θ (based on Eq. (1) and (2), replacing N_u by N_X) ;
    round θ to the nearest multiple of 0.25;
    use the resulting value of θ to announce;
  end
  $N_o = N_X - 1$;
end

Algorithm 1: Adjusted Transmission Probability

IV. Simulation and Results

In this section, we present the validation of our algorithm from Section III through simulations in the discrete event network simulator ns-3 [11]. The tool was used to implement D2D direct discovery type 1 according to 3GPP specifications [12] and our adaptive algorithm.

TABLE II: Scenarios

Scenario	Number of UEs	Optimal θ	Approximated θ
A	10	1	1.00
B	20	0.84703	0.75
C	40	0.46184	0.50
D	60	0.31644	0.25

TABLE III: Simulation Parameters and Values

Parameters	Values
UE transmission power	23 dBm
Propagation model	Friis, Cost231
Available bandwidth	50 RBs
Carrier frequency	700 MHz
Discovery period	0.32 s
Number of retransmission	0
Number of repetition	1
Number of resource block pairs	4
Number of subframes	5
Total number of resources	20
Total number of UEs	10, 20, 40, 60
Area Size	200 m × 200 m
Discovery start	2 s
Total simulations per scenario	100

A. Assumptions

We examined different UEs group sizes varying from 10 to 60 UEs while fixing the resource pool configuration, consisting of 4 resource block pairs ($N_f = 4$) and 5 subframes ($N_t = 5$), which provides a total of 20 discovery resources ($N_r = 20$).

Table II outlines the individual scenarios based on the UE populations used, their optimal transmission probabilities θ and the corresponding transmission probability values allowed by 3GPP. Table III summarizes a list of simulation parameters and their default values.

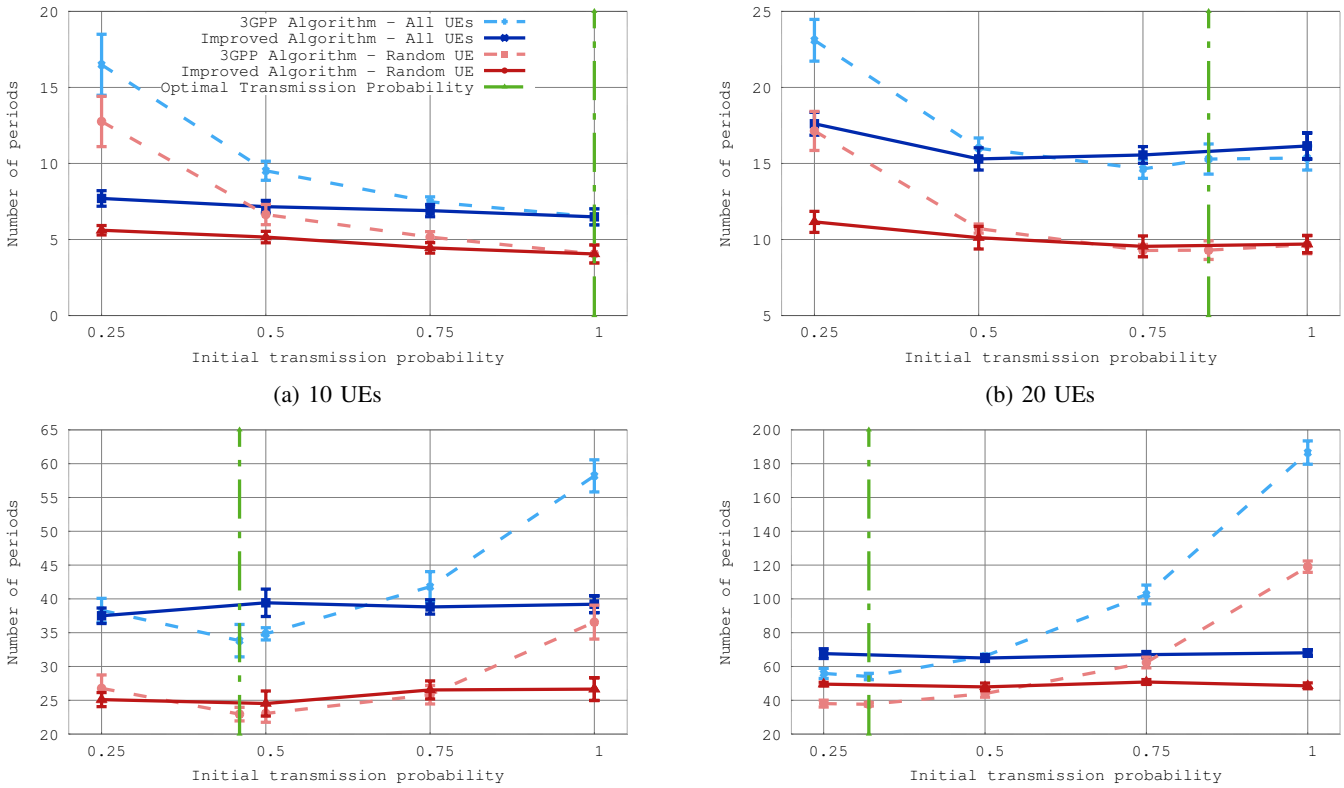
In each simulation, every UE is able to send announcements using a randomly chosen discovery resource. Users were deployed using a uniform random distribution within an area of 200 m × 200 m, ensuring that all UEs are within range of each other and therefore every UE can discover all other UEs in the group. We are aware that this doesn't address the hidden nodes problem, but the focus of this paper is on efficient mechanisms for a faster direct discovery.

We compare our adaptive algorithm to the standard 3GPP algorithm. Initially, we set the transmission probability to a defined value. As the 3GPP discovery algorithm is static, this pre-configured value will be used for the whole simulation when that algorithm is used. However, when using our proposed algorithm, all UEs will compute and update the value of their own transmission probability over time.

B. Stationary Topology

We assumed that all UEs are stationary. We were interested in computing and evaluating the time (measured in number of periods) required for all UEs in the group to discover each other, and the time required for one random UE to discover everyone else. We started our validation process with a baseline configuration where we discard all colliding discovery messages, and we used a simple propagation model with minimal propagation errors as assumed in [10].

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2



(a) 10 UEs

(b) 20 UEs

(c) 40 UEs

(d) 60 UEs

Fig. 1: Stationary Topology: Number of periods needed for all UEs to discover all other UEs in the group and for one random UE to discover everyone else in the group (Baseline)

We ran simulations for all four scenarios from Table II. Fig. 1 represents the corresponding averaged results, along with a confidence interval of 95 %.

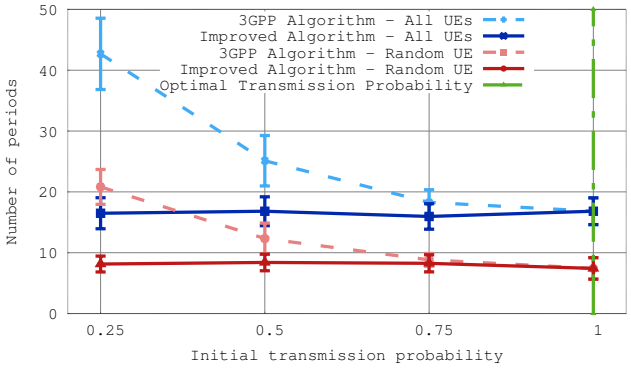
We observe that, in most cases, our algorithm (represented by the solid lines) outperforms the 3GPP algorithm (represented by the dashed lines), and only performs slightly worse when the transmission probability is configured with the optimal value from the start. The results for the number of periods needed for all UEs to discover all other UEs show trends similar to the results for the number of periods needed for one random UE to discover everyone else. In addition, the line corresponding to the adaptive algorithm performance fits a flat plot, which means that, independently of the initial transmission probability used, the number of periods needed to complete discovery is roughly the same.

For each scenario from Table II we can see how, without prior knowledge of the size of the group, there is a 25 % chance of starting the discovery process with the optimal transmission probability value. For that case, the 3GPP discovery algorithm would present better results given that the discovery process uses, since the beginning, the optimal configuration. However, we showed that the adaptive algorithm succeeded to perform similarly. For the other 75 % of the possible cases, using the pre-configured transmission probability, the UEs take longer to discover each other using a static algorithm. Our adaptive algorithm allows the UEs to complete the discovery faster, independently of their initial transmission probability, with the performance increase being significant in some cases. As we can see in Fig. 2d, using a transmission probability of 1 makes the 3GPP discovery take twice as long as our adaptive algorithm. This is due to our algorithm succeeding to detect the presence of a large number of UEs in the vicinity and adapting the transmission probability to the optimal value. We also note that, by the end of the discovery process, all UEs end up using the same transmission probability value. Therefore, the adaptive algorithm helps the UEs converge to the optimal θ, which means that future changes to the groups (e.g. new UEs arriving) will be discovered more efficiently, as the UEs are already carrying out the discovery process with an optimal configuration. This statement will be explored in Section IV-C, when a dynamic topology is considered.

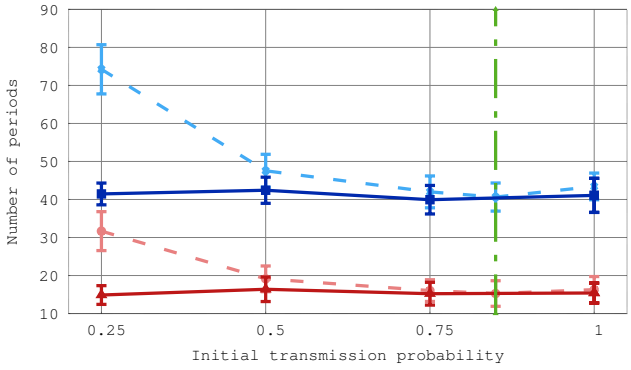
Once we have obtained promising results with the baseline configuration, we need to evaluate the performance when the channel is not ideal. For this purpose, we modify the previous configuration to represent a more realistic environment by using the propagation model “cost231” [13] and we try to retrieve at most one discovery message when there are collisions. Results of this Loss and Recovery configuration, along with a confidence interval of 95 %, are shown in Fig. 2.

As we can see, we obtained homogeneous plots with

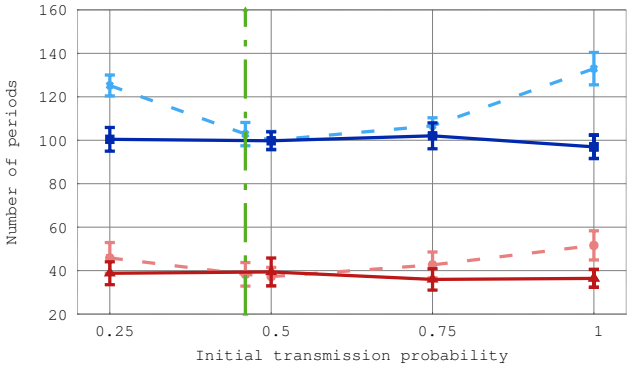
This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2



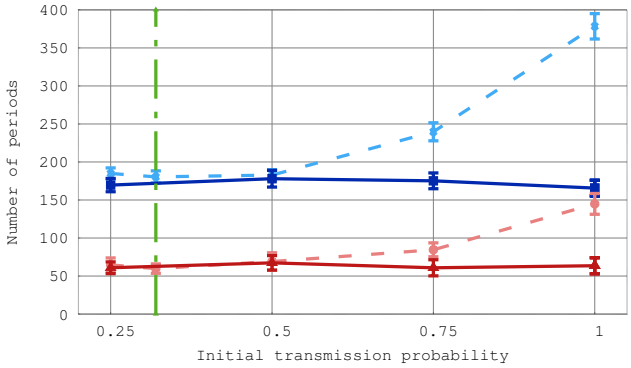
(a) 10 UEs



(b) 20 UEs



(c) 40 UEs



(d) 60 UEs

Fig. 2: Stationary Topology: Number of periods needed for all UEs to discover all other UEs in the group and for one random UE to discover everyone else in the group (Loss and Recovery)

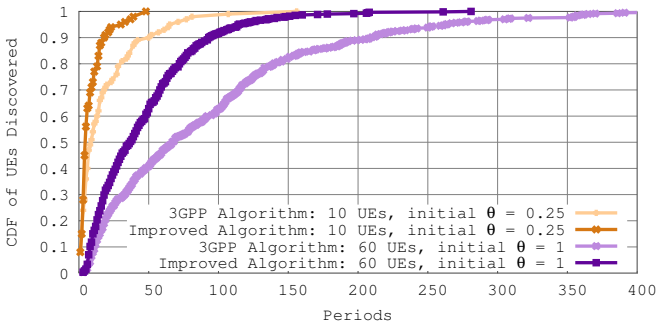


Fig. 3: Stationary Topology: CDF of UEs discovered in the group versus number of periods (Loss and Recovery)

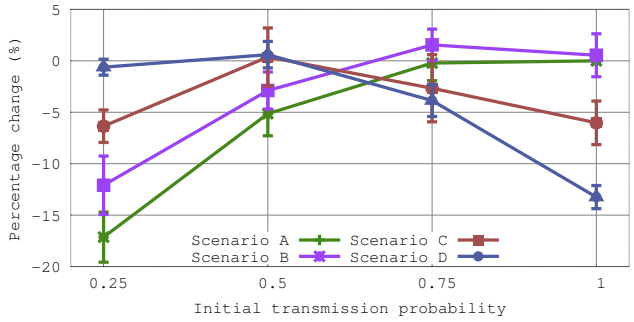
conclusions similar to those of the baseline scenario regarding the algorithm performance and matched curves. As expected, using a more stringent propagation model means that the discovery takes longer, even though the recovery process manages to save some of the announcements. We notice that the theoretical optimal value of the transmission probability is still valid even when using a more realistic error model.

Another way of showcasing the difference that our algorithm makes in the performance of the discovery process is to plot the Cumulative Distribution Function (CDF) of UEs discovered over time for two specific cases. We compared the number of UEs discovered in the group using the 3GPP algorithm and our adaptive algorithm. Fig. 3 shows the plots for scenario A using an initial transmission probability equal to 0.25, and scenario D for an initial transmission probability equal to 1. We have validated that the rest of the cases also show similar behaviors, with the distance between the curves being proportional to the differences shown in Fig. 2.

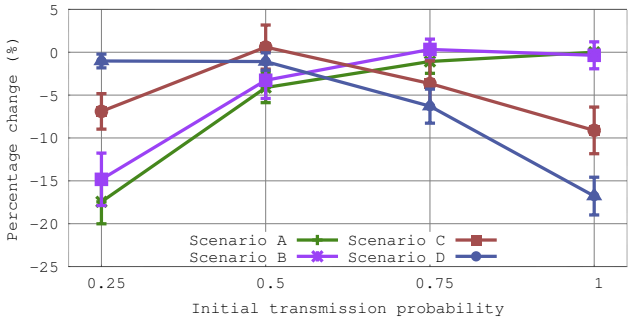
As we can see, our algorithm provides a significantly faster discovery. For example, for scenario D, when using the 3GPP algorithm, 95 % of the UEs are discovered in 263 periods. However, this value is reduced by more than half (115 periods) when our algorithm is applied, which is a significant improvement in the overall performance.

C. Dynamic Topology

To further evaluate our adaptive algorithm, we assume that one UE, noted UE_n, is joining the discovery group later on. Using its preconfigured transmission probability, UE_n will initiate discovery after the discovery process has already been completed for the other UEs. We verified that, for validation and testing purposes, the introduction of this additional UE does not change the approximate value of the optimal transmission probability in each scenario, mentioned in Table II, despite increasing the number of UEs in each group.



(a) for UE_n



(b) for all UEs in the group, including UE_n

Fig. 4: Dynamic Topology: Percentage change of the number of periods needed to complete discovery

This scenario allows us to evaluate the effect of new arrivals on the UEs' convergence to the optimal transmission probability. We are interested in assessing the time for UE_n to discover the rest of the UEs in the group, and the time for other UEs to detect UE_n's presence. In Fig. 4, we compute the percentage change (comparing the performance of our algorithm to 3GPP's) for UE_n (or all UEs, including UE_n) discovering the rest of the group (or all other UEs, respectively). A confidence interval of 95 % is computed.

When we use the optimal value of the transmission probability since the beginning of the simulations, the difference can be positive (percentage increase) but close to zero. But it is negative (percentage decrease) when using other initial transmission probabilities (i.e., 75 % of the possible cases), which means a reduction in the maximum number of periods needed to complete discovery compared to the 3GPP standard.

In scenario B, the optimal transmission probability is equal to 0.75. If the discovery starts using that value as its initial transmission probability, we record for Fig. 4a an increase of less than 2 % of the time needed to complete the whole discovery process. This only constitutes the worst case. The best registered amelioration is displayed in both Fig. 4a and 4b, for scenario A, when starting with an initial transmission probability equal to 0.25. Our algorithm helped to reduce the maximum number of periods needed by more than 17 %.

Overall, those results show significant improvement, by just taking into account the number of UEs discovered to adjust the transmission probability. There is little cost associated with our simple but efficient proposal. Our adaptive algorithm outperforms the 3GPP algorithm, even in situations where we have UEs joining the discovery group at a later time. It adjusts dynamically to a growing topology, thanks to its adaptive nature.

V. Conclusion and Future Work

In this paper, we proposed a novel adaptive algorithm that allows UEs to improve the performance of the discovery process in UE-Selected mode for LTE D2D. It dynamically adjusts the transmission probability based on vicinity awareness. The efficiency of our proposal was validated with simulations and the results showed that our algorithm reduces the time required for the discovery process in a group of UEs for several configurations. Furthermore, we have shown how our proposal also makes the discovery process perform better when changes in the topology happen.

This contribution opens up several new possibilities for future studies, such as detection of UE departure, and tuning the algorithm for more dynamic scenarios with groups of UEs causing bulk arrivals to and departures from the discovery group.

References

[1] 3GPP, "Feasibility Study for Proximity Services (ProSe)," 3rd Generation Partnership Project (3GPP), TR 22.803, 2013.

[2] 3GPP, "Proximity-based services (ProSe); Stage 2," 3rd Generation Partnership Project (3GPP), TS 23.303, 2015.

[3] 3GPP, "Study on LTE device to device proximity services; Radio aspects," 3rd Generation Partnership Project (3GPP), TR 36.843, 2015.

[4] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC)," 3rd Generation Partnership Project (3GPP), TS 36.331, 2015.

[5] S. Sun, Q. Gao, W. Chen, R. Zhao, and Y. Peng, "Recent progress of long-term evolution device-to-device in third-generation partnership project standardisation," *IET Communications*, vol. 9, no. 3, pp. 412–420, 2015.

[6] S. Xu and K. S. Kwak, "Network Assisted Device Discovery for D2D Underlying LTE-Advanced Networks," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, May 2014, pp. 1–5.

[7] K. W. Choi and Z. Han, "Device-to-Device Discovery for Proximity-Based Service in LTE-Advanced System," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 1, pp. 55–66, Jan 2015.

[8] T.-H. Ngo and Y. Kim, "Using Timing Advance to support proximity discovery in network-assisted D2D communication," in *2015 Seventh International Conference on Ubiquitous and Future Networks*, July 2015, pp. 926–928.

[9] D. Li and Y. Liu, "Performance analysis for LTE-A device-to-device discovery," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*. IEEE, 2015, pp. 1531–1535.

[10] D. Griffith and F. Lyons, "Optimizing the UE Transmission Probability for D2D Direct Discovery," in *IEEE Global Telecommunications Conference (GLOBECOM 2016)*, Washington D.C., USA, Dec 2016.

[11] NS-3 Documentation, "LTE Module in NS-3," accessed 08-February-2017. [Online]. Available: https://www.nsnam.org/docs/models/html/lte.html

[12] R. Rouil, F. J. Cintrón, A. Ben Mosbah, and S. Gamboa, "An LTE Device-to-Device module for ns-3," 2016, accessed 08-February-2017. [Online]. Available: https://www.nsnam.org/workshops/wns3-2016/posters/wns3_2016_LTE_D2D_NIST.pdf

[13] Commission of the European Communities, "Digital Mobile Radio Towards Future Generation Systems : Final Report," Luxembourg, 1999, accessed 08-February-2017. [Online]. Available: https://goo.gl/P06OZ7

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Physical Sidelink Control Channel (PSCCH) in Mode 2: Performance Analysis

David W. Griffith, Fernando J. Cintrón, Richard A. Rouil
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA
Email: {david.griffith, fernando.cintron, richard.rouil}@nist.gov

***Abstract*—User Equipments (UEs) that send data must advertise the upcoming transmission by broadcasting signaling messages over the Physical Sidelink Control Channel (PSCCH). Thus, it is important for the network operator to define the PSCCH resource pool to maximize the probability that each UE will be able to successfully decode all of the control messages that appear on the PSCCH. For UEs operating in Mode 2 (i.e., outside the coverage area of an eNodeB), this is especially challenging because there is no base station present that can assign PSCCH resources. UEs must choose pool resources randomly, which can lead to collisions of transmitted messages. In addition, UEs are half-duplex and a poorly designed control channel resource pool can create a significant risk that a signaling message and its duplicate will be missed by a UE that transmits its own signaling message in the same pair of subframes. In this paper, we present an analytical model that allows us to develop closed form expressions for the distribution of the number of UEs that successfully receive a transmitted message on the PSCCH. This model can support PSCCH design by network operators, and can be used to investigate other aspects of D2D communications.**

I. Introduction

The 3rd Generation Partnership Project (3GPP) introduced Proximity Services (ProSe) for Long Term Evolution (LTE) in Release 12 [1]. ProSe enables direct discovery of nearby UEs, and direct communication over a sidelink (rather than the cellular uplink or downlink) without relying on an evolved Node-B (eNB) to do coordination. While direct discovery is envisioned for the general public, direct communication is enabled for public safety users only.

Public safety users require Device-to-Device (D2D) communications when cellular coverage is not available, i.e., out-of-coverage. Out-of-coverage scenarios include remote areas lacking network infrastructure, inside buildings with deep fades, and during service outages. ProSe allows network operators to configure UEs to operate out-of-coverage, which is defined as Mode 2 [2]. Unfortunately, the lack of coordination between devices can degrade the communication performance.

In Mode 2, UEs must contend for resources in all sidelink channels. This paper develops a model of contention effects in the Physical Sidelink Control Channel (PSCCH), which carries signaling traffic that is vital for D2D data communications. We provide an overview of D2D communications in Section II, and derive the distribution of the number of UEs that receive a message over the PSCCH in Section III. In Section IV, we validate the model and discuss the network operator guidelines for PSCCH design that the model provides. We summarize our results in Section V.

Disclaimer: Certain commercial products are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the commercial products identified are necessarily the best available for the purpose.

II. Device-to-Device Communication

A. Overview and Prior Work

Communication over the sidelink uses communication periods that are periodic in the time domain. Each sidelink period includes instances of the PSCCH and the Physical Sidelink Shared Channel (PSSCH), which carries data. 3GPP defines procedures related to the transmission and reception on the PSCCH and PSSCH in [3, Clause 14]. All UEs are pre-configured by the network operator with the period duration, PSCCH configuration, and PSSCH configuration so that they can operate autonomously when out-of-coverage. Instead of using PSCCH resources that are assigned by an eNB, UEs with data to send select random resources from the PSCCH resource pool to send a control message that tells potential receivers, all of which monitor the PSCCH, where and how the transmitting UE's pending data will be transmitted in the PSSCH. Upon successful reception of a control message, a UE can tune to the indicated Physical Resource Blocks (PRBs) in the PSSCH. Any UEs that fail to receive and decode the PSCCH message will not be able to receive the advertised data transmission in the PSSCH.

A general description of the PSCCH and other D2D control channels is available from Lien et al. in [4] and [5]. A more detailed analysis by Shih et al. in [6] describes the PSCCH and PSSCH and derives a simple expression for PSCCH transmission success. The formula matches Eq. (20) in this paper, but does not include the half-duplex effect.

B. Physical Sidelink Control Channel

All control messages in a given period are sent twice; an initial transmission that occupies one PRB is followed by a duplicate transmission in a second PRB in the same period. UEs randomly select PRB pairs from the PSCCH resource pool, which is defined by the following pair of parameters: L_{PSCCH}, the number of subframes that the pool spans in the time domain ($2 \leq L_{PSCCH} \leq 40$), and $M_{RB}^{PSCCH\_RP}$, the number of PRBs that the pool spans in the frequency domain

Cintron, Fernando; Griffith, David; Rouil, Richard.
"Physical Sidelink Control Channel (PSCCH) in Mode 2: Performance Analysis."
Paper presented at 2017 IEEE International Conference on Communications (ICC), Paris, France. May 21, 2017 - May 25, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

[3, Clause 14.2.3]. Because each message transmission uses two PRBs, the number of available resources in the pool is $N_{PSCCH} = L_{PSCCH} \times \lfloor M_{RB}^{PSCCH\_RP}/2 \rfloor$.

III. Analytical Model

In this section, we develop the mathematical model that describes the ability of UEs in a D2D group containing N_u UEs to receive messages on the PSCCH. Each UE broadcasts a message over the PSCCH to all other UEs in the group. Let $\mathcal{R}_n^C$ be the event, "n UEs out of $(N_u - 1)$ UEs receive a message from a random UE in the group," where $0 \le n \le N_u - 1$. Note that when $\mathcal{R}_n^C$ occurs, at most n UEs will receive the data that the transmitting UE will send on the PSSCH. We will derive the distribution of $\mathcal{R}_n^C$ in this section.

A. Operation of the PSCCH

UEs with data to send randomly pick a resource from the pool and use the pair of PRBs associated with that resource to send a transmission advertisement. The resource index n_{PSCCH} is thus a discrete uniform random variable, $0 \le n_{PSCCH} < N_{PSCCH}$, and the mapping from n_{PSCCH} to the PRB pair occupying subframe ℓ_{b1} and PRB m_{a1}, and subframe ℓ_{b2} and PRB m_{a2}, where $0 \le a1, a2 < L_{PSCCH}$ and $0 \le b1, b2 < M_{RB}^{PSCCH\_RP}$ is (see [3, Clause 14.2.1.1])

$$a1 = \lfloor n_{PSCCH}/L_{PSCCH} \rfloor \tag{1a}$$

$$b1 = n_{PSCCH} \bmod L_{PSCCH} \tag{1b}$$

and

$$a2 = \left\lfloor \frac{n_{PSCCH}}{L_{PSCCH}} \right\rfloor + \left\lfloor \frac{M_{RB}^{PSCCH\_RP}}{2} \right\rfloor \tag{2a}$$

$$b2 = \left(n_{PSCCH} + 1 + \left\lfloor \frac{n_{PSCCH}}{L_{PSCCH}} \right\rfloor \bmod (L_{PSCCH} - 1) \right) \bmod L_{PSCCH}. \tag{2b}$$

We show an example of the resource mapping in Eqs. (1) and (2) in Fig. 1. As the figure shows, Eq. (1) fills the lower half of the pool, starting with 0 in the lower left hand corner and filling the pool by proceeding from left to right and from the bottom row to the top row of the lower half. Eq. (2) fills the upper half of the pool by starting in the next-to-leftmost subframe in the bottom row of the upper half, and filling from left to right, then wrapping around to the left of the current row before proceeding to the next higher row. The starting subframe in row $a2$ is one to the right from the starting subframe in row $(a2 - 1)$, and increasing resource indexes wrap around to the left, as shown in the figure.

We define $L \stackrel{\text{def}}{=} L_{PSCCH}$, $M \stackrel{\text{def}}{=} M_{RB}^{PSCCH\_RP}$, and $N \stackrel{\text{def}}{=} N_{PSCCH}$ in order to make the mathematical expressions in the following more compact. We assume that all UEs are half-duplex; i.e., two UEs that transmit in the same subframe will not receive each other's messages that were sent during that subframe.

Since a UE successfully decodes a PSCCH message if it can receive at least one transmission, we need to determine which resource selections by a UE will prevent it from receiving another UE's transmission. We begin by examining an arbitrary pair of UEs in the D2D group that we call the receiving UE and the transmitting UE. These two UEs independently choose resources. One of the following events will occur:

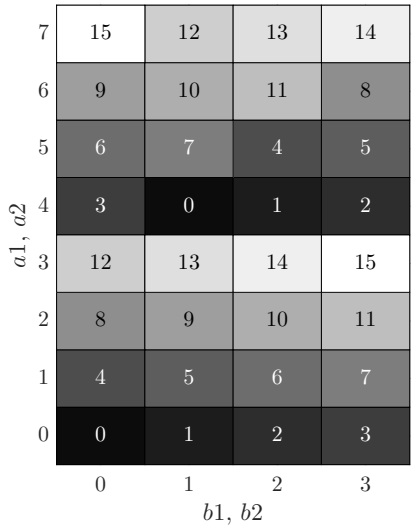


Fig. 1: Example PSCCH pool with $L_{PSCCH} = 4$ subframes and $M_{RB}^{PSCCH\_RP} = 8$ PRBs, with n_{PSCCH} values shown in each PRB.

- $\mathcal{X}$ (Collision): The transmitting and receiving UEs choose the same resource.
- $\mathcal{O}_2$ (Two column overlap): The transmitting and receiving UEs choose resources such that each of their respective PRBs lie in the same subframe (note that $\mathcal{X} \subseteq \mathcal{O}_2$).
- $\mathcal{O}_1$ (One column overlap): The transmitting and receiving UEs choose different resources, such that one of the transmitter's PRBs is in the same subframe as one of the receiver's PRBs, while the UEs' other PRBs lie in different subframes.
- $\mathcal{O}_0$ (Zero column overlap): The transmitting and receiving UEs choose different resources, such that neither UE's PRBs overlaps the other UE's PRBs.

In Fig. 2, we show examples of the each of the events described above. In each case shown in the figure, the transmitting UE has chosen $n_{PSCCH} = 0$. We show the Venn diagram associated with events $\mathcal{O}_0$, $\mathcal{O}_1$, and $\mathcal{O}_2$ in Fig. 3. We observe that the events are non-intersecting, and that the collision event $\mathcal{X}$ is a special case of event $\mathcal{O}_2$. This allows us to partition $\mathcal{O}_2$ into events $\mathcal{O}_2 \cap \mathcal{X}$ and $\mathcal{O}_2 \cap \overline{\mathcal{X}}$, where $\overline{\mathcal{X}}$ is the complement of event $\mathcal{X}$.

If two UEs' transmissions overlap in both of their chosen subframes, then event $\mathcal{X} \subseteq \mathcal{O}_2$ occurs if the two UEs pick the same resource, which results in a collision. Also, we assume that a collision blocks the transmitter UE's message for all UEs in the group[1]. Event $\mathcal{O}_2 \cap \mathcal{X}$ occurs if the

[1]In practice, one colliding message may still be received if the signal-to-interference ratio (SIR) is high enough. We assume that this does not occur, which simplifies the analysis and gives us a more conservative model. Our ongoing work will consider the effect of SIR.

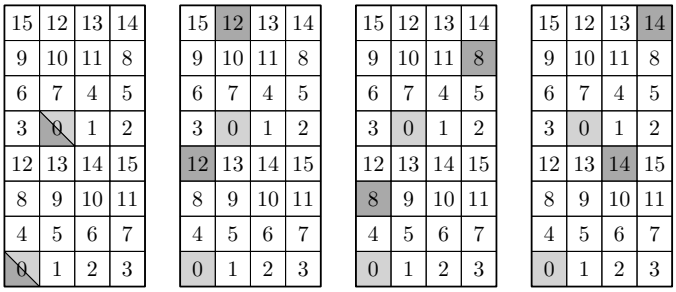


Fig. 2: Examples of UE interaction events $\mathcal{X}$, $\mathcal{O}_0$, $\mathcal{O}_1$, and $\mathcal{O}_2$.

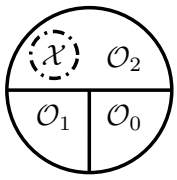


Fig. 3: Venn diagram for interactions between the resource selections by two UEs (dashed lines indicate that $\mathcal{X} \subseteq \mathcal{O}_2$).

receiver picks a resource that is different from the transmitter's resource, but which maps to a pair of PRBs that lie in the same two subframes as the PRB pair used by the transmitter. The half-duplex effect prevents the receiver from receiving either copy of the transmitter's message. If events $\mathcal{O}_0$ or $\mathcal{O}_1$ occur, a receiver will be able to receive at least one copy of the transmitter's message, assuming no other UEs pick the transmitter's chosen resource and cause a collision.

We condition on whether at least one of the $(N_u - 1)$ UEs picks the transmitter's resource index and causes a collision. We define $\mathcal{C}$ to be the event, "a collision occurred between the transmitter and at least one other UE." The complement $\overline{\mathcal{C}}$ is be the event that no collisions occurred between the transmitter and any other UE. We have

$$\begin{aligned}\mathsf{Pr}\{\mathcal{R}_n^C\} &= \mathsf{Pr}\{\mathcal{R}_n^C \,|\, \mathcal{C}\}\mathsf{Pr}\{\mathcal{C}\} + \mathsf{Pr}\{\mathcal{R}_n^C \,|\, \overline{\mathcal{C}}\}\mathsf{Pr}\{\overline{\mathcal{C}}\} \\ &= \left[1 - \left(1 - \tfrac{1}{N}\right)^{N_u - 1}\right]\delta[n] \\ &\quad + \left(1 - \tfrac{1}{N}\right)^{N_u - 1} \mathsf{Pr}\{\mathcal{R}_n^C \,|\, \overline{\mathcal{C}}\}, \end{aligned} \tag{3}$$

where $\delta[n]$ is the (discrete) Kronecker delta function:

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}.$$

To obtain Eq. (3), we observe that $\mathsf{Pr}\{\overline{\mathcal{C}}\} = \left(1 - \frac{1}{N}\right)^{N_u - 1}$ because the UEs pick resources independently, and a single receiving UE picks the transmitter's resource with probability $1/N$. Thus $\mathsf{Pr}\{\overline{\mathcal{C}}\}$ is the probability that none of the $(N_u - 1)$ receiving UEs pick the transmitter's resource. Next, $\mathsf{Pr}\{\mathcal{R}_n^C \,|\, \mathcal{C}\} = 0$ when $n > 0$, since a collision prevents any UE from receiving either copy of the transmitting UE's message. Thus for $n = 0$, $\mathsf{Pr}\{\mathcal{R}_0^C \,|\, \mathcal{C}\} = 1$, and $\mathsf{Pr}\{\mathcal{R}_n^C\}$ contains an additive term that is present only when $n = 0$. Finally, $\mathsf{Pr}\{\mathcal{R}_n^C \,|\, \overline{\mathcal{C}}\} \neq 0$ when $n = 0$, since it is possible for all the other UEs to pick resources that are different from the one picked by the transmitting UE, but that their resources can map to PRBs that lie in the same pair of subframes as those used by the transmitting UE.

B. Computing $\mathsf{Pr}\{\mathcal{R}_n^C \,|\, \overline{\mathcal{C}}\}$

In order to get $\mathsf{Pr}\{\mathcal{R}_n^C \,|\, \overline{\mathcal{C}}\}$, we can treat the resource selection process as a set of independent trials, each of which results in one of two possible outcomes. A trial succeeds if events $\mathcal{O}_0$ or $\mathcal{O}_1$ occur; it fails if event $\mathcal{O}_2$ occurs. From the form of Eq. (3), the success and failure probabilities for each receiving UE are conditioned on no collision with the transmitting UE, and are $\mathsf{Pr}\{\mathcal{O}_0 \cup \mathcal{O}_1 \,|\, \overline{\mathcal{X}}\}$ and $\mathsf{Pr}\{\mathcal{O}_2 \,|\, \overline{\mathcal{X}}\}$, respectively. From the Venn diagram in Fig. 3, the conditional success probability is

$$\mathsf{Pr}\{\mathcal{O}_0 \cup \mathcal{O}_1 \,|\, \overline{\mathcal{X}}\} = \frac{\mathsf{Pr}\{\mathcal{O}_0 \cup \mathcal{O}_1\}}{\mathsf{Pr}\{\overline{\mathcal{X}}\}} = \frac{\mathsf{Pr}\{\mathcal{O}_0\} + \mathsf{Pr}\{\mathcal{O}_1\}}{\mathsf{Pr}\{\overline{\mathcal{X}}\}}, \tag{4}$$

since $\mathcal{O}_0 \cup \mathcal{O}_1 \subseteq \overline{\mathcal{X}}$, and $\mathcal{O}_0 \cap \mathcal{O}_1 = \varnothing$. Similarly, the conditional failure probability is

$$\mathsf{Pr}\{\mathcal{O}_2 \,|\, \overline{\mathcal{X}}\} = \mathsf{Pr}\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\}/\mathsf{Pr}\{\overline{\mathcal{X}}\}. \tag{5}$$

Since $\mathsf{Pr}\{\mathcal{R}_n^C \,|\, \overline{\mathcal{C}}\}$ follows a binomial distribution, we use the success probability from Eq. (4) and the failure probability from Eq. (5) and get

$$\begin{aligned}&\mathsf{Pr}\{\mathcal{R}_n^C \,|\, \overline{\mathcal{C}}\} \\ &\quad = \binom{N_u - 1}{n} \frac{\left(\mathsf{Pr}\{\mathcal{O}_0 \cup \mathcal{O}_1\}\right)^n \left(\mathsf{Pr}\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\}\right)^{N_u - n - 1}}{\left(1 - \frac{1}{N}\right)^{N_u - 1}}. \end{aligned} \tag{6}$$

Inserting Eq. (6) into Eq. (3) and canceling the common term $\left(1 - \frac{1}{N}\right)^{N_u - 1}$ gives

$$\begin{aligned}&\mathsf{Pr}\{\mathcal{R}_n^C\} \\ &\quad = \left[1 - \left(1 - \tfrac{1}{N}\right)^{N_u - 1}\right]\delta[n] \\ &\qquad + \binom{N_u - 1}{n} \left(\mathsf{Pr}\{\mathcal{O}_0 \cup \mathcal{O}_1\}\right)^n \left(\mathsf{Pr}\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\}\right)^{N_u - n - 1}. \end{aligned} \tag{7}$$

We evaluate Eq. (7) by considering the structure of the resource pool. As shown in Fig. 4 for the case where $L_{PSCCH} = 4$ subframes and $M_{RB}^{PSCCH\_RP} = 8$ PRBs, one can show using Eqs. (1) and (2) that in general the PSCCH resource mapping is rotationally invariant. In other words, for all PRBs with a given PRB index m_{a1}, the position of the PRB (ℓ_{b2}, m_{a2}) relative to the PRB (ℓ_{b1}, m_{a1}) is the same, modulo $L_{PSCCH} - 1$. The fortuitous effect of this property is that we do not have to condition on which subframe is occupied by the transmitter's first transmission.

From Figs. 1 and 4, for each subframe, the copies of the resources in the lower (upper) half of the subframe are arranged in the same repeating pattern in the upper (lower) half of the subframe, as shown in Fig. 5. By partitioning the first subframe into two $(M/2) \times 1$ vectors, which are shaded differently in Fig. 4, we can see that the copies of the elements of the lower vector appear as diagonals that occupy the upper

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

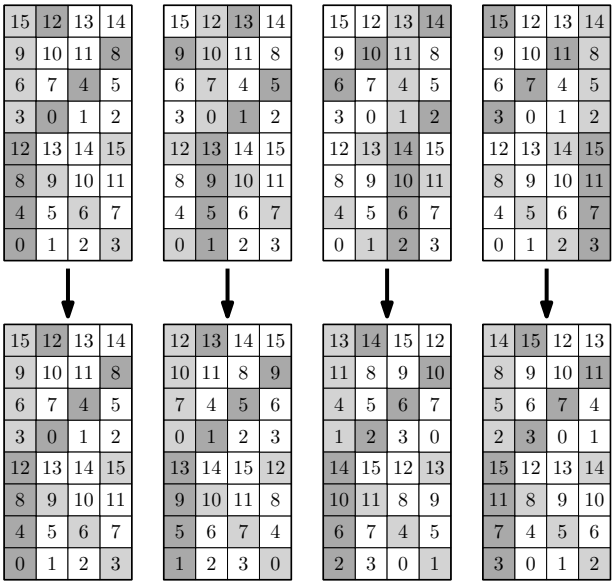
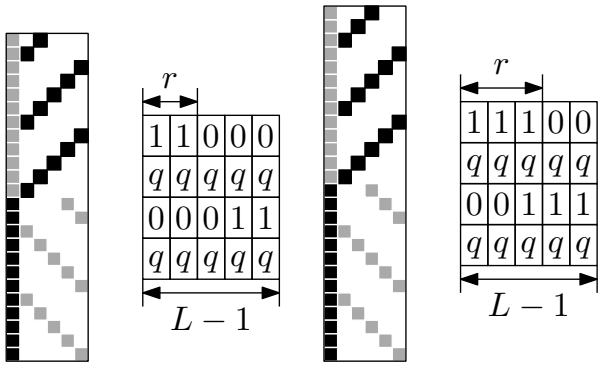


Fig. 4: Example of rotational invariance in the PSCCH, for $L_{PSCCH} = 4$ subframes and $M_{RB}^{PSCCH\_RP} = 8$ PRBs.



(a) Case 1: $0 \le r \le \left\lfloor \frac{L-1}{2} \right\rfloor$ (b) Case 2: $\left\lfloor \frac{L-1}{2} \right\rfloor < r \le L-2$

Fig. 5: Plots of example PSCCH pools that illustrate the two cases determined by the value of r.

rows of the pool and that the elements of the lower vector, if read from bottom to top, fill the diagonals in an ascending fashion, going from left to right. Likewise, the elements of the upper vector fill diagonals in an ascending fashion from right to left in the lower half of the pool.

We can map the diagonal patterns to a $4\times(L-1)$ occupancy grid, shown in Fig. 5. Let "Subframe 0" be the leftmost subframe. We define the following two quantities:

$$q \stackrel{\text{def}}{=} \lfloor (M/2)/(L-1) \rfloor \tag{8}$$

$$r \stackrel{\text{def}}{=} \frac{M}{2} \bmod (L-1), \tag{9}$$

so that

$$\frac{M}{2} = (L-1)q + r. \tag{10}$$

From Fig. 5, the second copies of the elements of each length-$(M/2)$ subvector in Subframe 0 form q complete diagonals with r elements forming a partial final diagonal. The final diagonal in the upper part of the PSCCH starts in Subframe 1 and extends rightward, while the final diagonal in the lower part of the PSCCH starts in Subframe $(L-1)$ and extends leftward.

There are two cases to consider, which depend on the value of r. Case 1 occurs when $0 \le r \le \lfloor (L-1)/2 \rfloor$; in this case, the partial diagonals do not overlap, as shown in Fig. 5a. Note that when $r = 0$ we have a set of q complete diagonals that appear over Subframes 1 through $(L-1)$, with no partial diagonal present. Case 2 occurs when $\lfloor (L-1)/2 \rfloor < r < L-1$; here, the partial diagonals overlap, as shown in Fig. 5b.

From Fig. 5a, for Case 1, $2r$ subframes contain $(2q+1)$ messages that are duplicates of messages in Subframe 0; the remaining $(L-2r-1)$ subframes each contain $2q$ duplicate messages from Subframe 0. For Case 2, there are $2(L-r-1)$ subframes containing $(2q+1)$ duplicate messages from Subframe 0, and the remaining $(2r-L+1)$ subframes each contain $(2q+2)$ duplicate messages from Subframe 0. We will use these facts in the following to obtain the conditional success and failure probabilities that compose Eq. (7).

C. Deriving $\mathsf{Pr}\{\mathcal{O}_0\}$

Let n_τ and n_ρ be the resource indexes respectively chosen by the transmitting and receiving UEs. We rotate the PSCCH grid so that one of the transmitting UE's two PRBs is in the first column (subframe s_0). Next, we label the subframes in the pool from left to right as follows: $s_0, s_1, \ldots, s_{L-1}$, so that one of the transmitter's PRBs lies in s_0, i.e., $\{n_\tau \in s_0\}$. Let s_τ be the subframe where the second PRB associated with n_τ occurs. By definition, $\mathcal{O}_0 = \{n_\rho \notin s_0\} \cap \{n_\rho \notin s_\tau\}$. We condition on the subframe occupied by the second PRB associated with n_τ, and get

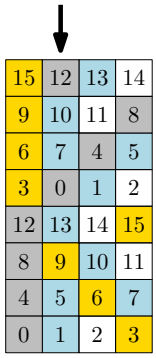
$$\mathsf{Pr}\{\mathcal{O}_0\} = \sum_{i=1}^{L-1} \mathsf{Pr}\{\mathcal{O}_0 \,|\, n_\tau \in s_i\}\, \mathsf{Pr}\{n_\tau \in s_i\}. \tag{11}$$

To evaluate Eq. (11), we examine $\mathsf{Pr}\{n_\tau \in s_i\}$. For Case 1, there are $2r$ subframes where $\mathsf{Pr}\{n_\tau \in s_i\} = (2q+1)/M$ and $(L-2r-1)$ subframes where $\mathsf{Pr}\{n_\tau \in s_i\} = (2q)/M$. For Case 2, there are $2(L-r-1)$ subframes where $\mathsf{Pr}\{n_\tau \in s_i\} = (2q+1)/M$ and $(2r-L+1)$ subframes where $\mathsf{Pr}\{n_\tau \in s_i\} = (2q+2)/M$.

Next, we consider $\mathsf{Pr}\{\mathcal{O}_0 \,|\, n_\tau \in s_i\}$. Fig. 6 shows an example where $n_\tau \in s_1$. Since there are no subframe overlaps, and $n_\tau \in s_0$, then $n_\rho \notin s_0$. In Fig. 6, $n_\tau \in \{0, 9, 12\}$; if the receiver chooses one of the other resources whose PRBs s_1, which are shaded light blue in Fig. 6, $\mathcal{O}_0$ cannot occur. The only choices that result in no overlaps are $n_\rho \in \{2, 11, 14\}$, The number of resources that the receiver can pick is N_{PSCCH} minus the number of resources in s_0 (M), minus the number of resources in s_1 that are not in s_0 (5 resources in the example). Thus $\mathsf{Pr}\{\mathcal{O}_0 \,|\, n_\tau \in s_1\} = (16-8-5)/16 = 3/16$.

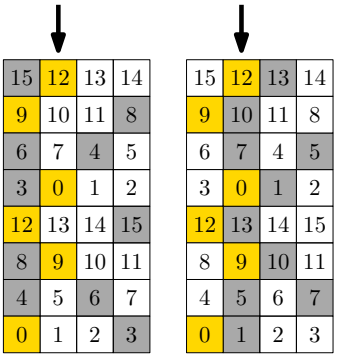
In general, for Case 1, there are $2r$ subframes where $\mathsf{Pr}\{\mathcal{O}_0 \,|\, n_\tau \in s_i\} = (N-2M+2q+1)/N$ and $(L-2r-1)$ subframes where $\mathsf{Pr}\{\mathcal{O}_0 \,|\, n_\tau \in s_i\} = (N-2M+2q)/N$. For Case 2, there are $2(L-r-1)$ subframes where $\mathsf{Pr}\{\mathcal{O}_0 \,|\, n_\tau \in s_i\} = (N-2M+2q+1)/N$ and $(2r-L+1)$ subframes where $\mathsf{Pr}\{\mathcal{O}_0 \,|\, n_\tau \in s_i\} = (N-2M+2q+2)/N$.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2



$$\Pr\{n_\rho \notin s_0 \cap n_\rho \notin s_1 \,|\, n_\tau \in s_1\} = \frac{16-8-(8-3)}{16} = \frac{16-2\times 8+3}{16} = \frac{3}{16}$$

Fig. 6: Example of possible double subframe overlaps when $n_\tau \in s_1$, for $L_{PSCCH} = 4$ subframes and $M_{RB}^{PSCCH\_RP} = 8$ PRBs.



$$\Pr\{n_\rho \in s_0 \cap n_\rho \notin s_1 \,|\, n_\tau \in s_1\} = \frac{2(8-3)}{16} = \frac{10}{16} = \frac{5}{8}$$

Fig. 7: Example of possible single subframe overlaps when $n_\tau \in s_1$, for $L_{PSCCH} = 4$ subframes and $M_{RB}^{PSCCH\_RP} = 8$ PRBs.

We now evaluate Eq. (11). We can simplify our result by replacing N with $LM/2$ and using Eq. (10). For Case 1,

$$\begin{aligned}\Pr\{\mathcal{O}_0\} &= (2r)\frac{(N-2M+2q+1)(2q+1)}{MN} \\ &\quad + (L-2r-1)\frac{(N-2M+2q)(2q)}{MN} \\ &= \frac{(L-4)M^2+4qM+4(2q+1)r}{LM^2},\end{aligned} \tag{12}$$

and for Case 2,

$$\begin{aligned}\Pr\{\mathcal{O}_0\} &= 2(L-r-1)\frac{(N-2M+2q+1)(2q+1)}{MN} \\ &\quad + (2r-L+1)\frac{(N-2M+2q+2)(2q+2)}{MN} \\ &= \frac{(L-4)M^2+4qM+4(2q+3)r-4(L-1)}{LM^2}.\end{aligned} \tag{13}$$

D. Deriving $\Pr\{\mathcal{O}_1\}$

Next, we consider a single subframe overlap. If we rotate the PSCCH so that $n_\rho \in s_0$, the event $\mathcal{O}_1$ occurs when $n_\rho \notin s_\tau$. Conditioning on the second subframe occupied by n_τ, we get

$$\Pr\{\mathcal{O}_1\} = \sum_{i=1}^{L-1} \Pr\{\mathcal{O}_1 \,|\, n_\tau \in s_i\} \Pr\{n_\tau \in s_i\}. \tag{14}$$

We obtained expressions for $\Pr\{n_\tau \in s_i\}$ in our treatment of $\Pr\{\mathcal{O}_0\}$. To get $\Pr\{\mathcal{O}_1 \,|\, n_\tau \in s_i\}$, we examine the example shown in Fig. 7. If $n_\rho \in s_0$ and $n_\tau \in s_1$ (i.e., if $n_\tau \in \{0, 9, 12\}$, which are shaded gold in the figure), then there are two possibilities: $n_\rho \in \{3, 4, 6, 8, 15\}$, or $n_\rho \in \{1, 5, 7, 10, 13\}$. Both sets of resources are shaded gray in the figure. The number of resources that the receiver can pick to get a single overlap is the number of gray resources in both grids, which is $M_{RB}^{PSCCH\_RP}$ minus the number of resources from s_0 that are in s_1. Thus the probability that the receiver picks a resource that results in event $\mathcal{O}_1$ is $2(8-3)/16 = 10/16$.

Generalizing this example, for Case 1, there are $2r$ subframes where $\Pr\{\mathcal{O}_1 \,|\, n_\tau \in s_i\} = (M-2q-1)/N$ and there are $(L-2r-1)$ subframes where $\Pr\{\mathcal{O}_1 \,|\, n_\tau \in s_i\} = (M-2q)/N$. For Case 2, there are $2(L-r-1)$ subframes where $\Pr\{\mathcal{O}_1 \,|\, n_\tau \in s_i\} = (M-2q-1)/N$ and $(2r-L+1)$ subframes where $\Pr\{\mathcal{O}_1 \,|\, n_\tau \in s_i\} = (M-2q-2)/N$.

Next we get the expressions for $\Pr\{\mathcal{O}_1\}$. For Case 1,

$$\begin{aligned}\Pr\{\mathcal{O}_1\} &= (2r)\frac{2(M-(2q+1))(2q+1)}{MN} \\ &\quad + (L-2r-1)\frac{2(M-2q)(2q)}{MN} \\ &= \frac{4M^2-8qM-8(2q+1)r}{LM^2},\end{aligned} \tag{15}$$

and for Case 2,

$$\begin{aligned}\Pr\{\mathcal{O}_1\} &= 2(L-r-1)\frac{2(M-2q-1)(2q+1)}{MN} \\ &\quad + (2r-L+1)\frac{2(M-2q-2)(2q+2)}{MN} \\ &= \frac{4M^2-8qM-8(2q+3)r+8(L-1)}{LM^2}.\end{aligned} \tag{16}$$

E. Deriving $\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\}$

Finally, we examine the case of double overlaps with no collision. Event $\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\}$ occurs when $n_\rho \in s_0$ and $n_\rho \in s_\tau$, but $n_\rho \neq n_\tau$. By conditioning on s_τ, we get

$$\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\} = \sum_{i=1}^{L-1} \Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}} \,|\, n_\tau \in s_i\} \Pr\{n_\tau \in s_i\}. \tag{17}$$

In the example shown in Fig. 8, $s_\tau = s_1$, so $n_\tau \in \{0, 9, 12\}$, which are shaded gold in the figure. There are three possibilities. If $n_\tau = 0$, $n_\rho \in \{9, 12\}$, if $n_\tau = 9$, $n_\rho \in \{0, 12\}$, and if $n_\tau = 12$, $n_\rho \in \{0, 9\}$. Note that $\Pr\{n_\tau = 0 \,|\, n_\tau \in s_1\} = \Pr\{n_\tau = 9 \,|\, n_\tau \in s_1\} = \Pr\{n_\tau = 12 \,|\, n_\tau \in s_1\} = 1/3$. As shown in the figure, this means that the conditional probability that $\mathcal{O}_2 \cap \overline{\mathcal{X}}$ holds is $3 \times (1/3) \times (2/16) = 1/8$.

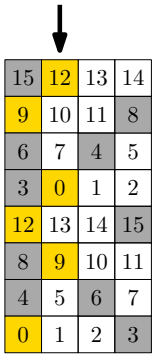
Generalizing this result, for Case 1, we have

- $2r$ subframes where $\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}} \,|\, n_\tau \in s_i\} = (2q)/N$
- $(L-2r-1)$ subframes where $\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}} \,|\, n_\tau \in s_i\} = (2q-1)/N$

For Case 2, we have

- $2(L-r-1)$ subframes where $\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}} \,|\, n_\tau \in s_i\} = (2q)/N$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2



$$\Pr\{\{n_\rho \in s_0\} \cap \{n_\rho \in s_1\} \cap \{n_\rho \neq n_\tau\} \,|\, n_\tau \in s_1\} = \left(\tfrac{1}{3} \times \tfrac{2}{16}\right) + \left(\tfrac{1}{3} \times \tfrac{2}{16}\right) + \left(\tfrac{1}{3} \times \tfrac{2}{16}\right) = \tfrac{1}{8}$$

Fig. 8: Example of double subframe overlap when $n_\tau \in s_1$, for $L_{PSCCH} = 4$ subframes and $M_{RB}^{PSCCH\_RP} = 8$ PRBs.

- $(2r - L + 1)$ subframes where $\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}} \,|\, n_\tau \in s_i\} = (2q+1)/N$

Thus for Case 1, we get

$$\begin{aligned}\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\} &= (2r)\frac{(2q)(2q+1)}{MN} \\ &\quad + (L - 2r - 1)\frac{(2q-1)(2q)}{MN} \\ &= \frac{(4q-2)M + 4(2q+1)r}{LM^2}, \end{aligned} \tag{18}$$

and for Case 2, we get

$$\begin{aligned}\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\} &= 2(L - r - 1)\frac{(2q)(2q+1)}{MN} \\ &\quad + (2r - L + 1)\frac{(2q+1)(2q+2)}{MN} \\ &= \frac{(4q-2)M + 4(2q+3)r - 4(L-1)}{LM^2}. \end{aligned} \tag{19}$$

We evaluate Eq. (7) by adding Eq. (12) to Eq. (15) (Case 1) or by adding Eq. (13) to Eq. (16) (Case 2) to get $\Pr\{\mathcal{O}_0 \cup \mathcal{O}_1\}$, and by using $\Pr\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\}$ from Eq. (18) (Case 1) or Eq. (19) (Case 2).

IV. Numerical Results and Evaluation

A. Validation

We used two independent methods to validate the analytical model: Monte Carlo experiments performed in Matlab, and network simulations performed using ns-3 [7]. We used a PSCCH pool configuration of 8 subframes by 44 PRBs, and a group size of 6 UEs. Our Monte Carlo experiments consisted of 100 experimental runs with 100 000 trials per run. The network simulations used the same transmission handling assumptions as the theoretical model: half duplex UEs and forced transmission drops resulting from collisions. Each data point used 50 simulation runs, each of which covered 8000 s of activity and used a sidelink communication period of 80 ms, so that we examined 100 000 periods per run.

In both the Matlab Monte Carlo runs and the ns-3 simulations, we randomly assigned resources to each UE, and then determined how many UEs each UE's message could reach, taking collisions and the half duplex effect into account. Once the full set of runs was complete, we obtained two sets

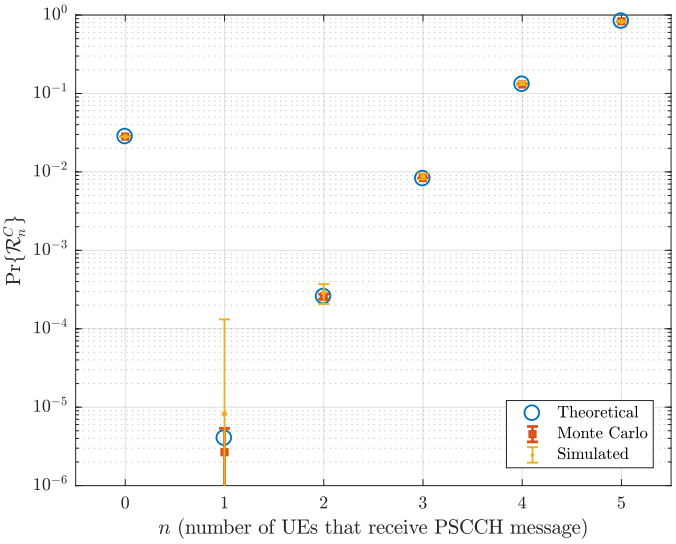


Fig. 9: Validation with 95 % confidence intervals; $N_u = 6$ UEs, $L_{PSCCH} = 8$ subframes, and $M_{RB}^{PSCCH\_RP} = 44$ PRBs.

of estimates of $\Pr\{\mathcal{R}_n^C\}$, one for the ns-3 simulations and one for the Monte Carlo simulations, by dividing $\mathcal{N}_n$, the number of times a UE's transmitted control message reached n other UEs, by the product $N_u \times \mathcal{N}_{\text{periods}}$. This results in one estimate per run. From each set of estimates, we computed estimates of both the mean $\widehat{\mu}_n$ and the standard deviation $\widehat{\sigma}_n$ for $0 \leq n \leq N_u - 1$; we plot $\widehat{\mu}_n$ in Fig. 9, and we used the estimated standard deviation to produce our 95 % confidence intervals that have the form $[\widehat{\mu}_n - 1.96\widehat{\sigma}_n, \widehat{\mu}_n + 1.96\widehat{\sigma}_n]$.

We present the theoretical results along with the outputs from both validation methods in Fig. 9, with 95 % confidence intervals shown for each simulation result. We note the large confidence interval at $n = 1$ for the results from ns-3 relative to the Monte Carlo results. This is because we used half as many runs in the ns-3 simulations due to time constraints. Moreover, $\mathcal{R}_1^C$ occurs only when $(N_u - 2)$ UEs do not collide with the transmitter but choose resources do that all experience double subframe overlaps, which is very unlikely, and which requires a large number of runs to produce any experimental occurrences of this event. We emphasize that for all values of n, the figure shows very close agreement between the model and the results obtained from both simulations. We obtained similar agreement using other pool configurations and other values of N_u; we do not present these results here due to space limitations.

B. PSCCH Pool Design

Finally, we show how that model can be used to produce design guidelines for network operators. Double overlaps are undesirable because they prevent UEs from receiving control messages, even in the absence of a collision. However, for a pool configured such that $L - 1 \geq M$, Case 1 as defined in Section III-B holds (i.e., $0 \leq r \leq \lfloor (L-1)/2 \rfloor$), and $q = 0$ and $r = M$. Evaluating Eq. (18) with $q = 0$ and $r = M$

gives $\mathsf{Pr}\{\mathcal{O}_2 \cap \overline{\mathcal{X}}\} = 0$. Evaluating Eq. (12) and Eq. (15) and summing the results gives $\mathsf{Pr}\{\mathcal{O}_0 \cup \mathcal{O}_1\} = 1 - (1/N)$, so that

$$\mathsf{Pr}\{\mathcal{R}_n^C\} = \begin{cases} 1 - \left(1 - \frac{1}{N}\right)^{N_u - 1}, & n = 0 \\ \left(1 - \frac{1}{N}\right)^{N_u - 1}, & n = N_u - 1 \\ 0, & \text{else} \end{cases} \quad (20)$$

Eq. (20) implies that there are two possible outcomes for a pool constructed to avoid double overlaps: a collision occurs that prevents a message from being received by any other UEs, or there is no collision and all UEs receive the transmitted message.

Fig. 10 shows the maximum D2D group size that is possible when $\mathsf{Pr}\{\mathcal{R}^C_{N_u-1}\} \geq 95$ % for various values of L_{PSCCH} and $M_{RB}^{PSCCH\_RP}$. The set of white discs follow the line $M_{RB}^{PSCCH\_RP} = L_{PSCCH} - 1$, and the regions of the figure that are shaded black indicate that it is not possible to support any UEs with the desired level of reliability. While the figure shows that increasing N_{PSCCH} increases the number of UEs that can be supported, it also shows that, for a given value of L_{PSCCH}, the maximum supportable group size is reached when $M_{RB}^{PSCCH\_RP} = L_{PSCCH} - 1$, and that increasing $M_{RB}^{PSCCH\_RP}$ does not produce further increases in the number of supportable UEs. In fact, in many cases the figure shows that further increasing $M_{RB}^{PSCCH\_RP}$ beyond L_{PSCCH} results in reductions in the maximum supportable group size. This is because expanding the pool in the frequency domain reduces $\mathsf{Pr}\{\mathcal{X}\}$, but increases $\mathsf{Pr}\{\mathcal{O}_2\}$.

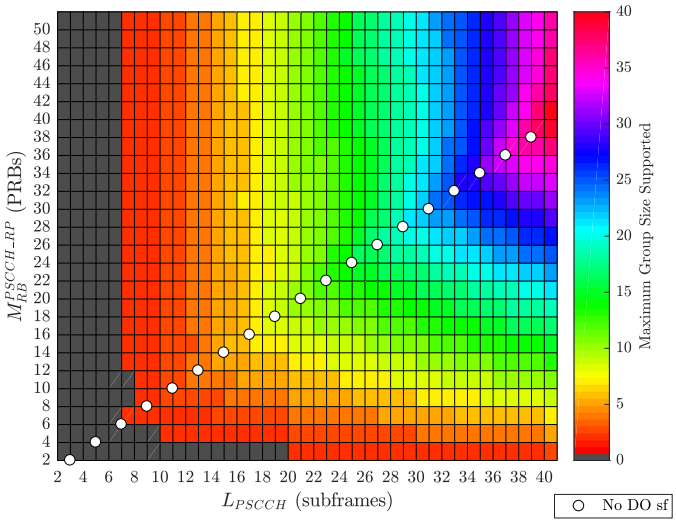


Fig. 10: Maximum D2D group size with $\mathsf{Pr}\{\mathcal{R}^C_{N_u-1}\} \geq 95$ % versus L_{PSCCH} and $M_{RB}^{PSCCH\_RP}$.

We also note that when $L_{PSCCH} \approx 8$ subframes, we can actually support very few UEs at the desired 95 % reliability level due to the effect of collisions. For example, when $L_{PSCCH} = 9$ subframes and $M_{RB}^{PSCCH\_RP} = 8$ PRBs, so that $N_{PSCCH} = 36$ resources, when $N_u = 4$ UEs, $\mathsf{Pr}\{\mathcal{X}\} = 1 - (35/36)^3 = 8.1$ %, which violates the minimum performance threshold. Were it possible for UEs to choose non-overlapping resources all the time, up to 36 UEs could be supported in this case with 100 % reliability, which is an order of magnitude increase. This suggests that basing resource index choices on the results of previous choices instead of using random selection may improve performance, which is a topic that we are investigating.

Thus, given $M_{RB}^{PSCCH\_RP}$, we can always improve the PSCCH performance by increasing L_{PSCCH}; however, a transmission period is split into the PSCCH and the PSSCH, so that increasing the size of the control pool requires reducing the duration of the PSSCH, which can decrease throughput. Therefore, the network operator must take any relevant constraints into account when designing the control pool.

V. Conclusion

In this paper, we developed an analytical model for the PSCCH in out-of-coverage scenarios where the UEs select control channel resources autonomously. We obtained the distribution of the number of UEs in a D2D group that receive a transmitted control message as a function of the PSCCH dimensions and the number of UEs in the group. This distribution can be used to generate performance metrics such as the maximum number of UEs that can be supported above a desired reliability threshold for a given resource pool configuration. Our analysis shows that the dimensioning of the resource pools has a significant and well-understood impact on the performance of the PSCCH. It is therefore possible to determine the best configuration to obtain a target performance level while minimizing the resource pool size. Finally, we note that the PSCCH is only one component of the sidelink. We are currently characterizing the performance of the PSSCH, and will use the results to examine how to optimize the transmission period to maximize sidelink throughput.

Acknowledgment

The authors would like to thank the Office for Interoperability and Compatibility of the United States Department of Homeland Security for funding this work.

References

[1] 3GPP, "Technical Specification Group Services and System Aspects; Proximity-based services (ProSe); Stage 2; TS 23.303," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2016. [Online]. Available: http://www.3gpp.org/DynaReport/23303.htm

[2] 3GPP, "Technical Specification Group Radio Access Network; Study on LTE Device to Device Proximity Services; Radio Aspects; TR 36.843," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2015. [Online]. Available: http://www.3gpp.org/DynaReport/36843.htm

[3] 3GPP, "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures; TS 36.213," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2016. [Online]. Available: http://www.3gpp.org/DynaReport/36213.htm

[4] S. Y. Lien, C. C. Chien, F. M. Tseng, and T. C. Ho, "3GPP device-to-device communications for beyond 4G cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 29–35, March 2016.

[5] S. Y. Lien, C. C. Chien, G. S. T. Liu, H. L. Tsai, R. Li, and Y. J. Wang, "Enhanced LTE device-to-device proximity services," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 174–182, December 2016.

[6] M. J. Shih, H. H. Liu, W. D. Shen, and H. Y. Wei, "UE autonomous resource selection for D2D communications: Explicit vs. implicit approaches," in *2016 IEEE Conf. Standards for Communications and Networking (CSCN)*, Berlin, Germany, 31 Oct–2 Nov 2016.

[7] R. A. Rouil, F. J. Cintrón, A. Ben Mosbah, and S. Gamboa Quintiliani, "A Long Term Evolution (LTE) device-to-device module for ns-3," in *Workshop on ns-3 (WNS3)*, Seattle, WA, USA, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3.5 GHz Environmental Sensing Capability Sensitivity Requirements and Deployment

Thao T. Nguyen, Anirudha Sahoo, Michael R. Souryal, and Timothy A. Hall
Communications Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, Maryland, U.S.
Email: {ttn1,ans9,souryal,tim.hall}@nist.gov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

***Abstract*—Spectrum sharing in the 3.5 GHz band between commercial and government users along U.S. coastal areas depends on an Environmental Sensing Capability (ESC), a network of radio frequency sensors and a decision system, to detect the presence of incumbent shipborne radar systems and trigger protective measures, as needed. It is well known that the sensitivity of these sensors depends on the aggregate interference generated by commercial systems to the incumbent radar receivers, but to date no comprehensive study has been made of the aggregate interference in realistic scenarios and its impact on the requirement for detection of the radar signal. This paper presents systematic methods for determining the required sensitivity and placement of ESC sensors to adequately protect incumbent shipborne radar systems from harmful interference. Using terrain-based propagation models and a population-based deployment model, the analysis finds the offshore distances at which protection must be triggered and relates these to a minimum required signal detection level at coastline sensors. We further show that sensor placement is a form of the well-known set cover problem, which has been shown to be NP-complete, and demonstrate practical solutions achieved with a greedy algorithm. Results show required sensitivities to be 4 dB to 16 dB lower than required by current industry standards. The methodology and results presented in this paper can be used by ESC operators for planning and deployment of sensors and by regulators for testing sensor performance.**

I. Introduction

The new Citizens Broadband Radio Service (CBRS) in the U.S. will share spectrum with government and non-government incumbents in the 3.5 GHz band. While initial use of the 3550 MHz to 3650 MHz portion of this band by CBRS will be restricted to geographic areas outside of coastal and certain inland exclusion zones, the CBRS rules and architecture allow for the eventual deployment of a sensing capability that will permit CBRS devices (CBSDs) to operate in these previously excluded zones. Termed an Environmental Sensing Capability (ESC), its purpose is to detect federal incumbent radar signals and communicate their presence (frequency and geographic area) to a Spectrum Access System (SAS) which coordinates CBSD access to the band. Together, the SAS and ESC must ensure that CBSDs do not generate harmful interference to incumbent systems.

The Federal Communications Commission has adopted rules for CBRS [1], and the Wireless Innovation Forum (WINNF) Spectrum Sharing Committee (SSC) is developing requirements and specifications for the SAS, ESC, and CBSDs. However, an open issue is the sensitivity requirement for ESC sensors, that is, the received signal level from an incumbent shipborne radar that a sensor must be able to detect to enable adequate interference protection. The sensitivity requirement is a function of the aggregate CBSD interference at the incumbent receiver: the greater the interference, the more sensitive a sensor needs to be. Preliminary work by the National Telecommunications and Information Administration (NTIA) and the WINNF SSC, discussed below, references the required sensitivity to the aggregate interference, but to our knowledge no study has quantified the aggregate interference and, thus, the sensitivity requirement.

This paper proposes a methodology for determining the ESC's required sensitivity as a function of CBSD deployment. Because realistic deployments of CBSDs will vary by population density and other factors, the proposed methodology takes these factors into account. Inputs include population data, terrain elevation, radio frequency transmitter and receiver characteristics, and incumbent interference protection criteria. We present formal algorithms both for determining a global sensitivity requirement and for determining the number and placement of ESC sensors from a set of candidate locations.

We formulate our sensor placement algorithm as the well-known *set cover* problem [2] and use a greedy algorithm to minimize the total number of ESC sensors required. Besides cost considerations, a key motivation for minimizing the number of sensors is to mitigate operational security concerns of the federal incumbent. However, there exists a tradeoff between minimizing the number of sensors for cost and security concerns on one hand and improving fault tolerance to sensor outage, on the other hand. To adjust this tradeoff, the algorithms proposed in this paper take as an additional input a *redundancy factor*, that is, the minimum number of ESC sensors that must simultaneously detect the incumbent. To illustrate their use and to provide representative results, the algorithms are applied to two specific coastal areas, with different population and terrain characteristics.

The paper is organized as follows. Section II reviews related work on ESC sensor sensitivity and placement, and compares them to the contributions of this paper. Section III describes the proposed methodology for determining sensor sensitivity and placement, including formal algorithm descriptions. Section IV details the modeling framework and assumptions,

U.S. Government work not protected by U.S. copyright

Hall, Timothy; Nguyen, Thao; Sahoo, Anirudha; Souryal, Michael.
"3.5 GHz Environmental Sensing Capability Sensitivity Requirements and Deployment."
Paper presented at IEEE Dynamic Spectrum Access Networks (DySPAN) Conference, Baltimore, MD. March 6, 2017 - March 9, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

including CBSD deployment, transmitter and receiver characteristics, and channel propagation models. Section V applies the methodology and modeling framework to two different coastal areas and presents the results. Finally, Section VI concludes with recommendations for future work.

II. Related Work

In a recently published technical report, NTIA proposed exploiting channel reciprocity to determine the received power level at which an ESC sensor must trigger detection [3]. The argument is based on the fact that the propagation loss from a shipborne radar transmitter to a sensor on the coastline is no more than the propagation loss from any land-based CBSD to the radar receiver. The analysis uses this principle to derive a trigger-detection threshold at the sensor of -64 dBm received radar peak power in a 1 MHz bandwidth. However, the derived threshold assumes a *single* co-channel Category B (high power) CBSD in the radar's beam. The authors recognized that, in practice, aggregate CBSD interference may exceed that of a single Category B CBSD, and that the trigger thresholds "will therefore be expected to vary along segments of coastline depending on the exact characteristics of actual CBSD deployments" [3, Section 2.6]. Our analysis in this paper seeks to address this very question, that is, to apply accepted models for CBSD deployment and channel propagation to predict the actual aggregate interference and thereby derive the trigger threshold along a segment of coastline.

The NTIA report goes on to propose a uniform ESC sensor spacing along the coastline of approximately 50 km, based on a geometric argument and the radio-horizon distance. The algorithms proposed in this paper, on the other hand, result in non-uniform sensor spacing that depends on CBSD deployment density, terrain features, and the desired sensor redundancy.

WINNF SSC requirements specify an ESC detection threshold in terms of the maximum propagation loss over which a coastline sensor must be able to detect shipborne radar [4]. The maximum loss is derived to be 184 dB in similar fashion to the trigger threshold of [3]: it corresponds to the path loss between a single coastline Category B CBSD and the radar receiver such that the interference-to-noise ratio (I/N) at the radar receiver is -6 dB. Similarly to the aforementioned NTIA report, the WINNF SSC requirements acknowledge that "ongoing investigations into the effects of aggregate interference from multiple CBSDs and their locations may need to be taken into account to better establish this figure" [4, Appendix A.2].

A straightforward methodology for uniform placement of ESC sensors is presented in [5]. Using a linear coastline with a parallel line in the water to represent the required radar detection distance, it finds the maximum distance between sensors that provides complete coverage of the detection distance along the coast. It presents a distance calculation for non-redundant coverage as well as one for redundant coverage. The differences in this approach and ours is that we use an actual map of the coastline and location-specific CBSD deployment densities that result in a non-uniform spacing of sensors.

The authors in [6] present an approach for optimal non-uniform sensor node placement. They use a piecewise linear representation of the coastline and of the interference contour. Possible sensor locations comprise a grid in the coastal land area. For each of the knot points (segment endpoints) in the interference contour, the grid point nearest in distance to it is selected as a candidate sensor location. This initial set is then used in a constrained optimization problem to find the minimum number of sensors needed to provide complete coverage of the knot points. They use a sequential convex programming algorithm to solve it. They also consider redundant sensor coverage. While the approach in [6] does result in non-uniform sensor placement, they use an abstract representation of the coast and interference boundary instead of using actual maps and modeling aggregate CBSD interference. A further distinction is that we identify the problem of sensor placement as a set cover problem and use a greedy algorithm to solve it.

The models used in our analysis are based in large part on those used by NTIA to develop the revised 3.5 GHz exclusion zones [7]. That study calculated the contours delineating areas in which CBSDs are not permitted to operate in the absence of an ESC, in order to protect federal incumbent radar systems. The NTIA study relied on a CBSD deployment model, CBSD and radar transmitter and receiver parameters, propagation models, and an aggregate interference model. Our analysis attempts to follow the models and assumptions in the NTIA exclusion zone analysis as closely as possible.

III. Overview of Approach

A. Problem Formulation

To analyze ESC sensor detection requirements, we divide the problem into three parts. First, the *interference contour* of a shipborne radar is computed. The interference contour defines the offshore boundary where the radar will begin to experience harmful interference when moving towards the coast. Second, the sensors' required sensitivity is determined such that a radar can be detected at any point on the interference contour. Finally, the third part is to determine the minimum number of ESC sensors and their locations such that a radar is detected with a desired level of redundancy. Minimizing the number of ESC sensors is desirable from an operational security standpoint in order to mitigate the risk of an adversary learning ship locations and frequencies, as well as to keep sensor deployment costs down.

To illustrate our approach, we apply our method to geographical areas around the two U.S. coastal cities of Virginia Beach and San Francisco. These regions differ both in population density and in terrain characteristics. To keep the computation manageable, we compute the interference contour as a piece-wise linear curve by connecting the discrete points at which maximum aggregate interference to the radar is just below the harmful interference threshold. Likewise, equally spaced discrete locations along the coast are chosen as *candidate* ESC sensor locations.

Now, we can formally state our problem as follows. Using the aggregate interference model, determine the piecewise linear interference contour as N discrete locations

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

$L_1, L_2, \ldots, L_N$. Let $E_1, E_2, \ldots, E_M$ be M discrete candidate locations for ESC sensor deployment. Let RD be the *redundancy factor* required for detection of the radar, i.e., at least RD number of ESC sensors should be able to simultaneously detect the radar when it is on the interference contour. Assuming that each ESC sensor will have the same sensitivity, compute the required sensitivity. The next task is to determine the minimum number of ESC sensors and their locations (among the M candidate locations) such that a radar at any of the N locations on the interference contour is detected with a redundancy factor of RD.

B. Computation of Interference Contour

The computation of the interference contour, presented in Algorithm 1, starts with initial ship locations along the coastline. Each initial ship location gives rise to one point on the interference contour. CBSDs are randomly deployed in proportion to the population density, and the maximum aggregate interference to the radar receiver (over all azimuth angles of the radar antenna) is computed at a given ship location. If the aggregate interference exceeds the harmful interference threshold (I_T), then the ship location is moved by a discrete step size away from the coast. The step-size iterations stop at the location where the aggregate interference falls below I_T. At this point, the algorithm checks if this location is indeed on (or near) the interference contour within a level of statistical confidence by repeating the last computation for 100 random, independent deployments of CBSDs (Lines 18 to 28).

C. Computation of Sensitivity of ESC Sensors

To compute the required sensitivity of the ESC sensors, Algorithm 2 first calculates the received peak power at each candidate ESC sensor location from a given radar location on the interference contour. It then picks the RD^{th} highest received power as the sensitivity that would ensure redundancy factor RD when detecting radar at that given location (Line 5). This process is repeated for every radar location, and the minimum among those sensitivity values is chosen as the required sensitivity for all ESC sensors.

D. Placement of ESC Sensors

The algorithm for placement of ESC sensors, Algorithm 3, starts with a *detection matrix*, whose rows represent candidate ESC sensor locations on the coast and whose columns represent radar locations on the interference contour. An entry is 1 if the ESC sensor at that row can detect, or *cover*, the radar located at that column; otherwise it is 0. Then, the problem reduces to choosing the minimum number of rows from the detection matrix which together can cover all the radar locations. This is the *set cover* problem, which is known to be NP-complete [2]. Hence, Algorithm 3 uses an iterative *greedy* method to find the set cover. In each iteration, the greedy method chooses, from the unselected rows, the row which has the maximum number of 1's at radar locations which are still not covered (Line 14). The greedy method also ensures that the redundancy factor is taken into account while computing the coverage of each radar location (Line 22).

TABLE I
SHIPBORNE RADAR-1 TECHNICAL PARAMETERS.

Radar-1 Parameter	**Value**
Transmitted Power to Antenna (dBm)	90
Mainbeam Antenna Gain (dBi)	32
Antenna Directivity/Patterns	Recommendation ITU-R M.1851
Half-power beamwidth (degree)	0.81
Transmit/Receive Bandwidth (MHz)	1
Center Frequency (MHz)	3600
Antenna Height (m)	50
Insertion/Cable Losses (dB)	2
Noise Figure (NF) (dB)	3
Interference-to-Noise Ratio (I/N) (dB)	-6

For finite lengths of coastline, the greedy algorithm may choose a candidate sensor location that is not necessary to satisfy the coverage requirement. For this reason, a final pruning step should be applied to remove such locations.

Antenna directionality may also affect sensor placement, due to the need to provide sufficient overlapping coverage of directional antenna patterns. However, the impact of directionality on placement was neglected in this analysis.

IV. ANALYSIS MODEL

This section describes the models and assumptions used in this analysis. They include propagation models, terrain and other databases, the aggregate interference model, the CBSD deployment model, and the technical parameters of the incumbent radar, CBSD, and ESC sensor. Wherever possible, the same models and assumptions used in [7] are used in this analysis.

A. Shipborne Radar Technical Parameters

The federal incumbent radar system is the one referred to as Shipborne Radar 1 in [7]. The technical parameters for the radar transmitter and receiver are obtained from [3], [7] and are summarized in Table I.

The generalized mathematical model of the radar system antenna is described in Recommendation ITU-R M.1851 [8]. It is used to obtain the radar receive and transmit antenna gain in the azimuth and elevation orientations in the direction of the CBSDs.

Given the radar receiver bandwidth and the noise figure, the receiver noise power can be computed as follows:

$$N = 10\log_{10}(k \times T \times BW_{rx} \times 10^6) + NF \qquad (1)$$

where N is the receiver noise power (dBm), $k = 1.38 \times 10^{-23}$ is Boltzmann's constant (J/K), T is the receiver temperature (K), BW_{rx} is the receiver bandwidth (MHz), and NF is the receiver noise figure (dB). If the receiver has a bandwidth of 1 MHz, 3 dB noise figure, and a temperature of 290 K, the receiver noise power is -111 dBm.

The interference threshold, I_T, at the radar receiver can be determined as:

$$I_T = N + I/N \qquad (2)$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Algorithm 1: Compute Interference Contour

Input: $l_1, l_2, \ldots, l_N$: Initial ship locations along the coastline
$I\_threshold$: Interference threshold that defines harmful interference to the radar receiver
stepSize: distance ship moves away from coast in every iteration
p_{th} : defines the number of maximum aggregate interference values, out of 100 iterations, which should be less than interference threshold to declare the corresponding ship location as a point on the interference contour
Output: $L_1, L_2, ..., L_N$: ship locations which define the piece-wise linear interference contour

for *each initial ship location* $l_i \in \{l_1, l_2, \ldots, l_N\}$ **do**
  $l_{tmp} = l_i$;
  $I_{agg}[1..360] = 0$;
  $done1\_iter = NOT\_DONE$;
  while *(done1_iter != DONE)* **do**
    randomly deploy CBSDs within the area around l_i ;
    for *each azimuth of the main-beam of the radar antenna* $az_j \in \{1, .., 360\}$ **do**
      $I_{agg}[j]$ =Aggregate interference from CBSDs to the radar receiver at location l_{tmp} ;
    $I\_MaxAgg\_az = \max_{1 \le j \le 360}(I_{agg}[j])$;
    if $I\_MaxAgg\_az > I\_threshold$ **then**
      l_{tmp} = new ship location after it is moved by $stepSize$ away from the coastline ;
    else
      $done1\_iter = DONE$;
  $I_{agg}[1..360] = 0$;
  $I\_MaxAgg[1..100] = 0$;
  $done100\_iter = NOT\_DONE$;
  while *(done100_iter != DONE)* **do**
    for *k = 1 to 100* **do**
      randomly deploy CBSDs within the area around l_i ;
      for *each azimuth of the main-beam of the radar antenna* $az_j \in \{1, .., 360\}$ **do**
        $I_{agg}[j]$ = Aggregate interference from CBSDs to the radar receiver at location l_{tmp} ;
      $I\_MaxAgg[k] = \max_{1 \le j \le 360}(I_{agg}[j])$;
    I_count = number of maximum aggregate interference values in $I\_MaxAgg[] \le I\_threshold$;
    if $I\_count \le p_{th}$ **then**
      l_{tmp} = new ship location after it is moved by $stepSize$ away from the coastline ;
    else
      $L_i = l_{tmp}$;
      $done100\_iter = DONE$;
return $\{L_1, L_2, .., L_N\}$;

Algorithm 2: Compute Sensitivity of ESC Sensor

Input: $L_1, L_2, \ldots, L_N$: radar locations on the interference contour
$E_1, E_2, \ldots, E_M$: candidate ESC sensor locations
RD: Redundancy Factor /* minimum number of ESC sensors required to simultaneously detect any given radar location */
Output: $\mathcal{S}$: required minimum sensitivity of each ESC sensor

for *each radar location* $L_j \in \{L_1, L_2, \ldots, L_N\}$ **do**
  for *each candidate ESC location* $E_i \in \{E_1, E_2, \ldots, E_M\}$ **do**
    $P_r[i]$ = Received peak power at E_i when radar is at location L_j ;
  Sort the elements of $P_r[\,]$ in a non-increasing order ;
  $S[j] = P_r[RD]$ /* 'RD' number of ESC sensors can detect radar at location L_j, when sensitivity of the ESC sensors is set at $S[j]$ */
$\mathcal{S} = \min_{1 \le j \le N}(S[j])$ /* Pick the minimum of the sensitivities */
return $\mathcal{S}$;

where I_T is the interference threshold (dBm), and I/N is the maximum permissible interference-to-noise ratio at the radar receiver (dB). If I/N is set to -6 dB as in [7], the interference threshold is -117 dBm.

B. ESC Sensor Technical Parameters

The ESC sensor technical parameters are provided in Table II. The power levels are referenced to the antenna input, therefore the antenna gain and antenna patterns of the ESC sensor are neglected.

TABLE II
ESC SENSOR TECHNICAL PARAMETERS.

ESC Sensor Parameter	Value
Receive Bandwidth (MHz)	1
Center Frequency (MHz)	3600
Antenna Height (m)	6
Insertion/Cable Losses (dB)	2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Algorithm 3: Placement of ESC Sensors

Input: $L_1, L_2, \ldots, L_N$: radar locations on the interference contour
$E_1, E_2, \ldots, E_M$: candidate ESC sensor locations
$\mathcal{S}$: sensitivity of each ESC sensor
RD: Redundancy Factor /* minimum number of ESC sensors required to simultaneously detect any given radar location */
Output: ESC_set : set of deployment locations of ESC sensors

Initialize Detection Matrix $D[1..M][1..N] = 0$;
for *each radar location* $L_j \in \{L_1, L_2, \ldots, L_N\}$ **do**
  for *each candidate ESC location* $E_i \in \{E_1, E_2, \ldots, E_M\}$ **do**
    P_r = Received peak power at E_i when radar is at location L_j ;
    if *(*$P_r \geq \mathcal{S}$*)* **then**
      $D[i][j] = 1$ /* ESC sensor at location E_i can detect radar at location L_j */

/* $D[\ ][\ ]$ is the detection matrix */
$placement = NOT\_DONE$; $ESC\_set = \emptyset$;
$covered[1..N] = 0$;
$Sensor\_loc[1..M] = NOT\_SELECTED$;
/* keeps track of ESC locations which are still available */
$Holes[1..N] = 0$;
/* keeps track of radar locations which are covered */
$COVERAGE\_MATRIX[1..M][1..N] = 0$;
/* start the greedy method */
while *(placement != DONE)* **do**
  Let $AVAIL\_LOC$ be the set of indices in $Sensor\_loc[\ ]$ whose values equal $NOT\_SELECTED$;
  Let $i \in AVAIL\_LOC$ be the index of the row in $D[\ ][\ ]$ which has maximum number of 1's at the positions corresponding to 0's in $Hole[\ ]$;
  In case of a tie, pick the row with maximum number of 1's in the entire row;
  $COVERAGE\_MATRIX[i] = D[i][1..N]$;
  /* copy the i^{th} row of D matrix; Sensor location at row i covers maximum radar locations not covered yet */
  $Holes[1..N]$ $\&= COVERAGE\_MATRIX[i][1..N]$ /* bit-wise AND the two vectors to mark the corresponding radar locations as covered */
  $Sensor\_loc[i] = SELECTED$; $ESC\_set = ESC\_set \cup \{E_i\}$;
  $covered[1..N] = covered[1..N] + COVERAGE\_MATRIX[i][1..N]$ /* vector addition */
  if *all elements in* $Holes[] == 1$ **then**
    for *each element* i *in* $covered[1..N]$ **do**
      if $covered[i] < RD$ **then**
        $Holes[i] = 0$ /* this radar location still needs coverage with required redundancy */
    if *all elements in* $Holes[] == 1$ **then**
      $placement = DONE$;
return ESC_set ;

C. Initial Ship and ESC Sensor Locations

Geographic Information System (GIS) 2011 National Land Cover Database (NLCD) data [9] is used to place the initial ship locations and the candidate ESC sensors along the coast in this analysis. The NLCD 2011 data is divided into 30 m by 30 m pixels and assigns a land cover classification code to each pixel (e.g., dense urban, urban, suburban, rural).

The initial ship locations are placed along the edge of the NLCD data, which is close to the shoreline. The separation between 2 ship locations is 333 pixels (i.e., approximately 10 km) in latitude. The locations of the candidate ESC sensors can be found by projecting the initial ship locations on the shoreline, which is formed along the open water regions with classification code of 11.

Fig. 1 illustrates the placement of initial ship and candidate sensor locations near Virginia Beach. The area of interest extends around 200 km along the coast. There are 19 initial ship locations $\{L_1, \ldots, L_{19}\}$ and 19 candidate sensor locations $\{E_1, \ldots, E_{19}\}$ placed along the coast.

While the candidate sensor locations are equally spaced in this example, our methods apply just as well to any arbitrary set of candidate locations. Thus, one can exclude areas that are unavailable for sensor deployment, such as certain private

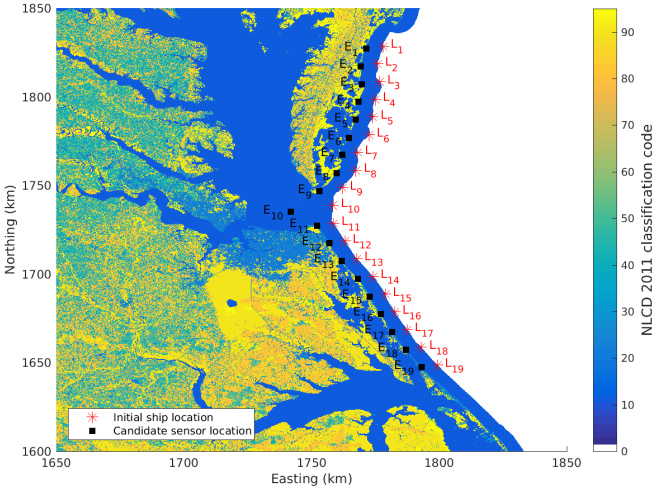


Fig. 1. Initial ship and candidate sensor locations.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

TABLE III
CBSD TECHNICAL PARAMETERS.

CBSD Parameter	Value	
EIRP (dBm)	30 (Outdoor), 26 (Indoor)	
Channel Bandwidth (MHz)	10	
Signal Bandwidth (MHz)	9	
Center Frequency (MHz)	3600	
Region	Channel Usage (%)	Percent Indoor
Dense Urban/Urban	60	80
Suburban	40	99
Rural	20	99
Antenna Height (m)		
Outdoor	6	
Indoor - Dense Urban	50%:3-15; 25%:18-30; 25%:33-60	
Indoor - Urban	50%: 3; 50%: 6-18	
Indoor - Suburban	70%: 3; 30%: 6-12	
Indoor - Rural	80%: 3; 20%: 6	
Building Atten. Loss (dB)	20%:20; 60%:15; 20%:10 (Indoor)	
Insertion/Cable Losses (dB)	2 (Outdoor)	

properties, wildlife refuges, etc.

D. CBSD Technical and Deployment Parameters

The CBSD technical and deployment parameters are listed in Table III. Only low-power Category A CBSDs are considered in line with the assumptions of [7], but the analysis can easily accommodate high-power Category B CBSDs, as well.

Four data sources are used to deploy the CBSDs within an area of interest, i.e., the NLCD 2011 data [9], the 2010 U.S. Census population data [10], the census tract polygons [11], and the daytime commuter factors [12], as well as other assumptions described in [7].

The pixels of the NLCD data are grouped into 90 m by 90 m bins. The classification of a bin (dense urban, urban, suburban, or rural) is determined by the majority of classification codes of its component pixels. As mentioned in [7], the number of CBSDs per classification is computed from the population density, the daytime traveling factor, a market penetration factor of 20 %, a channel scaling factor of 10 %, and a ratio of users to CBSD for each classification.

Fig. 2 illustrates an example of CBSD deployment extending 150 km west, north, and south, and 120 km east of the initial ship location L_{11}. Table IV shows the calculation of the number of CBSDs in detail; the "daytime population" includes the daytime commuter adjustment, MP is the market penetration factor, and CS is the channel scaling factor. In this example, the total number of CBSDs deployed in the area of interest is 6 772. The CBSDs are deployed randomly by varying different parameters including location, indoor antenna height, building attenuation, and clutter loss.

E. Propagation Models

Two propagation models, the ITS Irregular Terrain Model (ITM) and the extended Hata (eHATA) model, are used to compute the median basic transmission loss from the CBSD to the radar receiver. The point-to-point mode is used in both

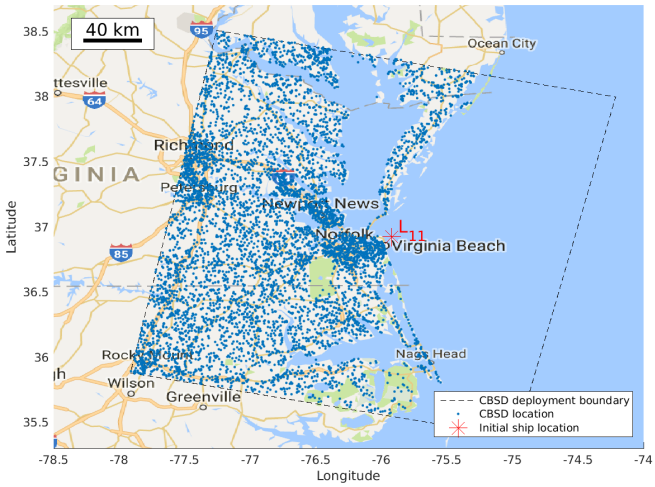


Fig. 2. Sample CBSD deployment.

TABLE IV
CALCULATION OF NUMBER OF CBSDS.

Region	Population	Daytime Popul.	MP	CS	CBSD/ User	No. of CBSDs
Urban	448 760	489 077	0.2	0.1	0.02	196
Suburban	775 051	809 581	0.2	0.1	0.05	810
Rural	867 118	864 844	0.2	0.1	0.33	5 766

models, and the great circle terrain elevation profile between the CBSD and radar location is extracted and used as input to these models. For CBSDs in dense urban, urban, and suburban environments with a height above ground of less than 18 m, the maximum of the ITM and eHATA basic transmission losses is used. For CBSDs in rural areas—as well as in dense urban, urban, and suburban areas above 18 m—only the ITM model is used. For rural CBSDs, an additional, random clutter loss, uniformly distributed in the range (0 to 15) dB, is applied.

For the path loss from the radar transmitter to the ESC sensor, only the ITM model is used based on the assumption that coastline ESC sensors are located in rural areas. In addition, no additional clutter loss is added to this median basic transmission loss.

F. Aggregate Interference Calculation

For each ship location, the azimuth angle of the radar antenna is swept 360 degrees in 1 degree increments. The aggregate interference from all CBSDs in the area is computed for each azimuth angle of the main beam of the radar antenna. The maximum aggregate interference over all azimuth angles is compared with the radar receiver's interference threshold.

a) Interference Calculation for a Single Path: The interference power received at the radar from each CBSD is computed as follows:

$$I = EIRP_{CBSD} - L_{i\_CBSD} - L_{building} - L_{prop} - L_{clutter} + G_{radar} - L_{i\_radar} - B_{radar/CBSD} \quad (3)$$

where I is the received interference power at the output of the radar antenna (dBm), $EIRP_{CBSD}$ is the equivalent isotropically radiated power (EIRP) from the CBSD (dBm), L_{i_CBSD}

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

is the CBSD transmitter insertion loss (dB), $L_{building}$ is the building attenuation loss (dB), L_{prop} is the median propagation loss from the CBSD transmitter to the radar receiver, $L_{clutter}$ is the clutter loss (dB), G_{radar} is the radar receiver antenna gain toward the CBSD (dBi), L_{i_radar} is the radar receiver insertion loss (dB), and $B_{radar/CBSD}$ is the frequency dependent rejection (dB).

The frequency dependent rejection is defined as $B_{radar/CBSD} = 10\log_{10}(B_{radar\_rx}/B_{CBSD\_tx})$, if $B_{radar\_rx} < B_{CBSD\_tx}$; and $B_{radar/CBSD} = 0$, otherwise. Note that B_{radar_rx} and B_{CBSD_tx} are the bandwidths of the radar receiver and the CBSD transmitter, respectively.

b) Aggregate Interference: Given the interference power computed for each individual path from the CBSD transmitter to the radar receiver, the aggregate interference power to the radar receiver is:

$$I_{agg} = 10\log\Big(\sum_{k=1}^{N} 10^{I_k/10}\Big) \tag{4}$$

where I_{agg} is the aggregate interference level at the radar receiver from all CBSD transmitters (dBm), N is the number of CBSD transmitters, and I_k is the interference power at the radar receiver from each individual CBSD transmitter (dBm).

G. Model Validation

We validated our implementation and use of the aforementioned CBSD deployment, propagation, and aggregate interference models by repeating the NTIA exclusion zone analysis [7] in selected coastal areas. In this analysis, CBSDs are randomly deployed as described in Section IV-D, and the aggregate interference is computed at a fixed ship location 10 km offshore. Specifically, the azimuth angle of the radar antenna is swept in one-degree increments, and at each angle the radial distance is determined at which the aggregate I/N at the radar receiver from CBSDs beyond that distance drops just below -6 dB. This process is repeated for 10 000 independent CBSD deployments. The exclusion zone boundary at a given azimuth angle is based on the 95$^{\text{th}}$ percentile of the radial distances obtained from the Monte Carlo iterations.

In one departure from [7], we used the United States Geological Survey (USGS) Digital Elevation Model terrain database [13] rather than the resampled Spatial Data Transfer Standard terrain data used by NTIA [14] due to incompatibility of the latter with our geodata software. However, both databases have the same resolution of 90 m (3-arc-second).

Fig. 3 shows our computed 95$^{\text{th}}$ percentile distances (the dark-green dashed line) overlaid on top of the results from [7] for an area near San Diego, California. The 95$^{\text{th}}$ percentile distances match very well with those of [7] for most azimuth angles, except between 305° and 350°. The discrepancy at these angles could be due to the usage of different terrain databases.

V. Analysis Results

We apply the methodology for determining the required ESC sensor sensitivity and placement to two U.S. coastal areas, San Francisco on the west coast and Virginia Beach on the east coast. The former has a higher population as well as a higher terrain elevation relative to sea, factors which are expected to lead to greater interference to shipborne radar.

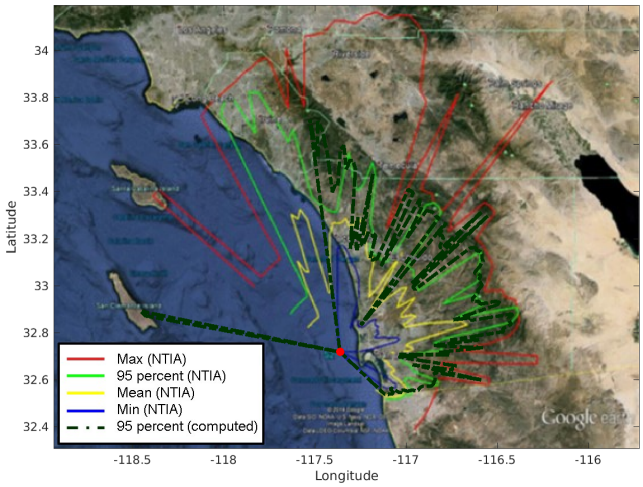


Fig. 3. Overlaid exclusion zone results.

The National Oceanic and Atmospheric Administration (NOAA) Global Land 1-km Base Elevation (GLOBE) terrain database [15] is used to extract the elevation profiles between the CBSD and radar and between the radar and ESC sensor. The GLOBE database has a coarser resolution, i.e., 1 km (30-arc-second), than other databases. It was used in this study to increase the speed of extracting the elevation profile.

A. Virginia Beach

Results for the interference contour as well as the sensitivity requirements and deployment of ESC sensors for Virginia Beach are described below.

1) Interference Contour: We applied Algorithm 1 developed in Section III to find the piece-wise linear curve along which the aggregate interference caused by CBSDs to the radar receiver is just below a permissible I/N of -6 dB. The initial ship locations were placed near the shoreline as described in Section IV and were moved with a step size of 10 km away from shore. The algorithm stopped when at least 95 of the 100 random CBSD deployments resulted in aggregate interference below the interference threshold (2). Fig. 4 shows the histogram of the aggregate interference at one of the locations on the interference contour. Across the 19 ship locations on the interference contour, the standard deviation of the interference ranged from 3 dB to 9 dB.

Fig. 5 shows the interference contour result for the Virginia Beach area. The distance from each point on the interference contour to the shoreline ranges from 36 km to 67 km. As expected, the distance depends on the number of CBSDs deployed in the surrounding area, i.e., the more CBSDs are deployed, the larger the interference distance needed to protect the incumbent.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

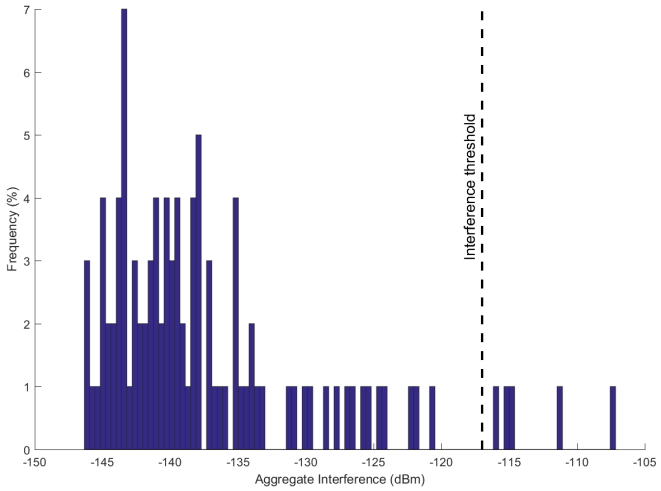


Fig. 4. Histogram of aggregate interference at point L_{11} on the Virginia Beach interference contour showing 95 % of realizations below the -117 dBm threshold.

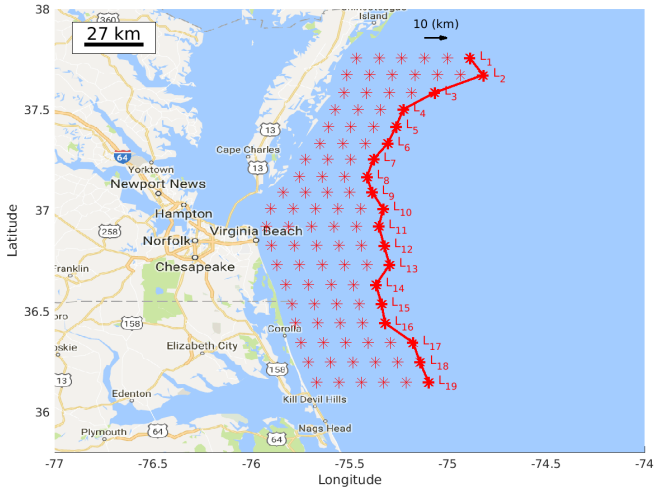


Fig. 5. Interference contour near Virginia Beach.

2) ESC Sensor Sensitivity Requirements and Deployment: We derive the sensitivity requirements and location of ESC sensors along the coast near Virginia Beach by applying Algorithms 2 and 3 as described in Section III, respectively.

For each ship location on the interference contour, we compute the received peak power from the radar at all 19 candidate ESC sensor locations along the coast using an equation similar to (3). The received peak power is measured when the main beam of the radar transmitter is pointed directly toward the sensor location. Fig. 6 shows the received peak power from ship location L_{11} to all candidate sensor locations $\{E_1, \ldots, E_{19}\}$.

a) Redundancy Factor = 1: If each ship location on the interference contour is required to be detected by only one ESC sensor, using Algorithm 2, the sensitivity requirement for the ESC sensor is computed to be -68 dBm/MHz. The corresponding detection matrix is computed and shown in Fig. 7.

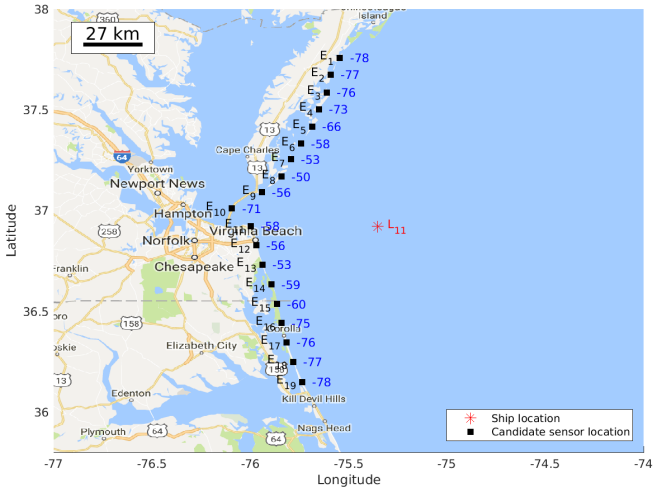


Fig. 6. Received peak power in dBm at ESC sensor locations.

Virginia Beach: RF = 1, Sensitivity = -68 dBm/MHz

Candidate sensor location \ Ship location	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_10	L_11	L_12	L_13	L_14	L_15	L_16	L_17	L_18	L_19
E_1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
E_2	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
E_3	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
E_4	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
E_5	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
E_6	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
E_7	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
E_8	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
E_9	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
E_10	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
E_11	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
E_12	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
E_13	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
E_14	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
E_15	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
E_16	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
E_17	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0
E_18	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
E_19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
No. det. sensors	1	1	2	2	2	2	2	1	1	1	2	1	1	2	2	2	2	1	1

Fig. 7. Detection matrix with redundancy factor of 1, Virginia Beach.

Applying Algorithm 3 to this detection matrix, a set of candidate sensor locations $\{E_1, E_5, E_{15}, E_{18}\}$ is selected. The sensor location E_5 is selected first since it can detect the most ship locations (9) on the interference contour. (Sensor locations E_7 and E_{13} also detect 9 interference locations, but E_5 is selected as it precedes them in the ordered list). The sensor location E_{15} is selected next because it can detect the most uncovered ship locations by E_5, i.e., $\{L_{12}, \ldots, L_{17}\}$, followed by E_1 covering $\{L_1, L_2\}$, and E_{18} covering $\{L_{18}, L_{19}\}$. The last row in the figure shows the number of selected sensor locations that can detect each ship location on the interference contour (the sum along each column of the highlighted rows).

Fig. 8 depicts the selected candidate sensor locations $\{E_1, E_5, E_{15}, E_{18}\}$ and their associated coverage of the interference contour. Each ship location can be detected by at least one selected sensor location. Ignoring the edge sensors, whose placement is affected by the finite length of coastline, we observe that the spacing between sensors E_5 and E_{15} is approximately 100 km.

For reference, the analysis in [3] found the required sensitivity to be -64 dBm/MHz and gave a uniform spacing between sensors of 50 km, while the analysis in [5] found the required

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

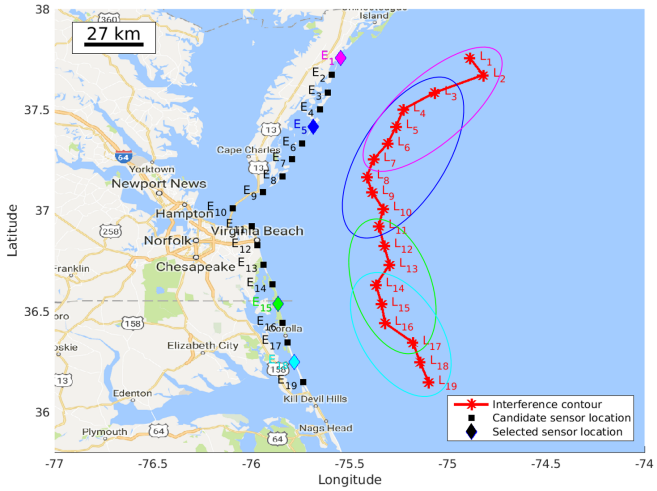


Fig. 8. Coverage of the interference contour (redundancy factor = 1), Virginia Beach.

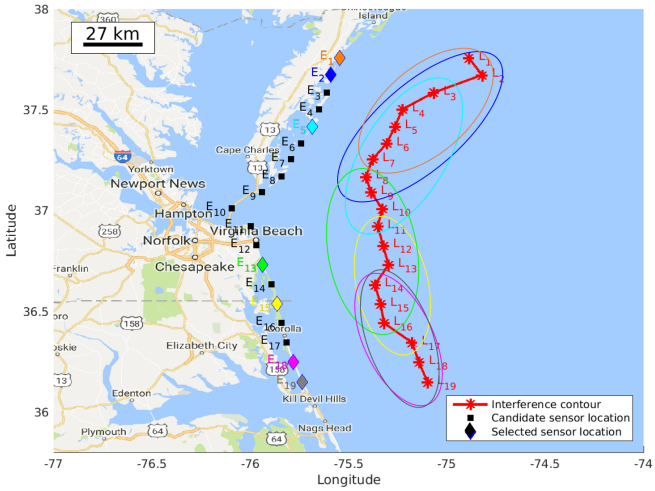


Fig. 9. Coverage of the interference contour (redundancy factor = 2), Virginia Beach.

sensitivity to be -100 dBm/MHz with a spacing of 264 km when the interference contour is 70 km from shore.

b) Redundancy Factor = 2: If each ship location on the interference contour is required to be detected by at least two ESC sensors, i.e., the redundancy factor is equal to 2, the sensitivity requirement for the ESC sensor lowers (becomes more sensitive) by 2 dB.

Applying Algorithm 3, the set of candidate sensor locations in the order of selection is $\{E_2, E_{13}, E_{18}, E_5, E_{15}, E_1, E_{19}\}$. With this selection, at least 2 selected sensor locations can detect any ship location on the interference contour.

Fig. 9 depicts the selected candidate sensor locations $\{E_2, E_{13}, E_{18}, E_5, E_{15}, E_1, E_{19}\}$ and their associated coverage of the interference contour. Compared to the previous case with redundancy factor of 1, nearly twice the number of sensors are needed. To reduce the number of sensors while still meeting the redundancy requirement, one could improve the sensitivity of the ESC sensor such that its coverage area increases. The tradeoff between sensor sensitivity and number of deployed sensors could be a subject of future work.

B. San Francisco

We repeat the same analysis for the San Francisco area, which is more densely populated and, in general, has higher terrain elevation than the Virginia Beach area. Of interest is to what extent a different environment affects the interference contour, sensitivity requirement, and sensor placement.

1) Interference Contour: There are 20 initial ship locations $\{L_1, \ldots, L_{20}\}$ and 20 candidate sensor locations $\{E_1, \ldots, E_{20}\}$ used in this area. The resulting interference contour from these initial ship locations is shown in Fig. 10. In this case, the offshore distance of the interference contour ranges from 106 km to 146 km, roughly two to three times that observed in the Virginia Beach area. This is due to the significantly larger number of CBSDs deployed at higher elevation in San Francisco as compared to Virginia Beach.

2) ESC Sensor Sensitivity Requirements and Deployment: Given the interference contour, the sensitivity requirement and set of sensor locations are determined for redundancy factors of 1 and 2, as follows.

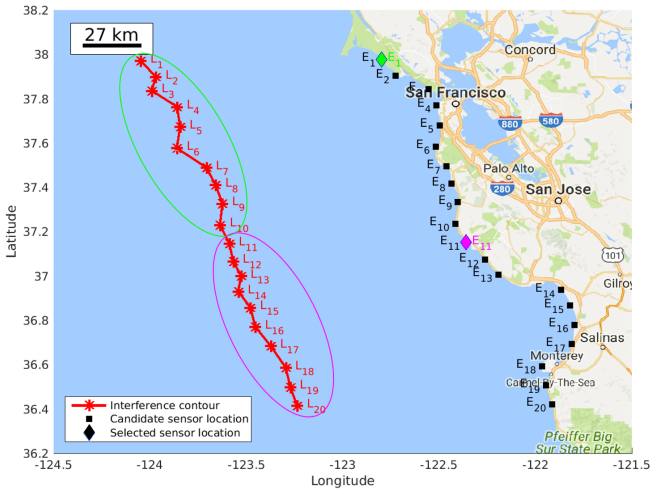


Fig. 10. Coverage of the interference contour (redundancy factor = 1), San Francisco.

a) Redundancy Factor = 1: For a redundancy factor of 1, the sensitivity requirement for the ESC sensor is found to be -80 dBm/MHz, 12 dB more sensitive than in Virginia Beach, and the set of sensor locations after pruning is found to be $\{E_1, E_{11}\}$. Fig. 10 depicts the selected sensor locations and their coverage of the interference contour.

b) Redundancy Factor = 2: For a redundancy factor of 2, the sensitivity requirement for the ESC sensor lowers (becomes more sensitive) by only 0.3 dB. The set of sensor locations in order of selection is $\{E_9, E_1, E_{11}, E_2\}$. Fig. 11 depicts the selected candidate sensor locations and their coverage of the interference contour. In this case, the greedy algorithm achieves the desired redundancy with twice the number of sensors.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

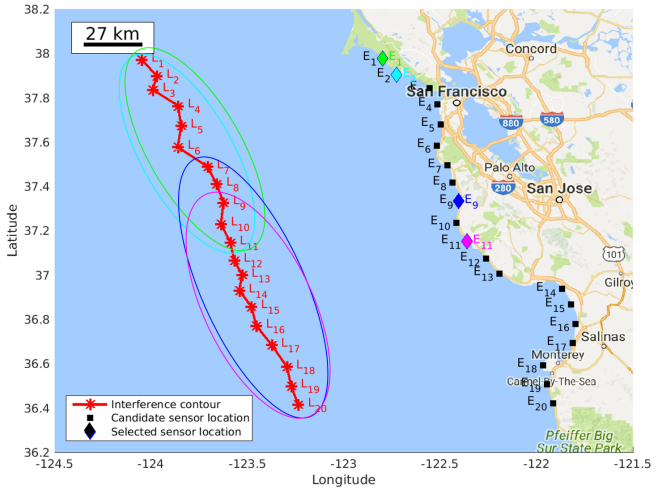


Fig. 11. Coverage of the interference contour (redundancy factor = 2), San Francisco.

VI. Conclusion

In summary, we have presented a methodology for determining the required sensitivity and placement of ESC sensors to adequately protect 3.5 GHz incumbent shipborne radars from harmful interference. Given a maximum allowable interference-to-noise ratio (I/N threshold) at the radar receiver, we described a systematic algorithm for determining the boundary at sea at which a shipborne radar would experience interference at this threshold. Termed the *interference contour*, this boundary depends on the aggregate interference at the radar receiver from CBSDs deployed on land, computed from propagation models using terrain elevation data as well as clutter and building attenuation losses. In illustrative examples near Virginia Beach and San Francisco, on the eastern and western coasts of the U.S., respectively, we found that the interference contour ranged from 36 km to 146 km offshore, depending on the number of CBSDs deployed in the surrounding area.

We also described systematic algorithms that, given the interference contour at sea, determine the locations of coastline sensors and their required sensitivity so that a radar crossing any point of the interference contour can be detected. We showed that the sensor selection algorithm is a form of the well-known set cover problem and applied a greedy approach to obtain solutions. The algorithm finds the minimum number of sensors to cover the interference contour with the desired level of sensor redundancy for fault tolerance. In the two examples provided, we found that a 200 km segment of coastline can be addressed by 2 to 4 sensors with a required sensitivity of -68 dBm/MHz to -80 dBm/MHz, in coastal areas with a lower and higher CBSD density, respectively. We note that these received signal levels are 4 dB to 16 dB lower than the detection thresholds proposed in [3], [4]. To achieve dual-sensor redundancy, roughly twice the number of sensors are needed.

In this work, we solved for sensor sensitivity and sensor placement as a two-step process. First, we found a global sensitivity requirement for all sensors along a segment of coastline, and then used that sensitivity in the placement algorithm. Future work should consider the *joint* selection of sensor sensitivity and placement, with the potential that some sensors are more sensitive than others. Naturally, this analysis can be applied to the U.S. coastlines in their entirety as follow-on work, as well.

Another consideration for future work is the signal-to-interference ratio at the ESC sensor. This paper only analyzed the received signal level at the sensor from the incumbent radar transmitter. However, in practice, sensors will also experience co-channel interference from CBSDs. While this interference can be mitigated to some extent with the use of directional antennas pointed to sea, irregular coastlines and nearby CBSD transmitters will nonetheless generate unwanted interference at the sensor. Hence, an important figure of merit for sensor detection performance is the signal-to-interference ratio (SIR). The analysis in this paper can be extended to predict the SIR at each sensor and to factor this metric into the sensor placement algorithm.

References

[1] "Citizens broadband radio service," 2 C.F.R. § 96, 2016.
[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
[3] F. H. Sanders, E. F. Drocella, and R. L. Sole, "Using on-shore detected radar signal power for interference protection of off-shore radar receivers," National Telecommunications and Information Administration, Technical Report TR 16-521, Mar. 2016. [Online]. Available: http://www.its.bldrdoc.gov/publications/2828.aspx
[4] "Requirements for commercial operation in the U.S. 3550–3700 MHz citizens broadband radio service band," Wireless Innovation Forum Document WINNF-15-S-0112, Version V1.0.0, May 2016. [Online]. Available: https://workspace.winnforum.org/kws/public/document?document_id=2413
[5] "Application of Google Inc. for Certification to Provide Spectrum Access System and Environmental Sensing Capability Services," GN Docket No. 15-319, Appendix B: Environmental Sensing Capability (ESC) Siting Considerations, 2016. [Online]. Available: https://ecfsapi.fcc.gov/file/60001851224.pdf
[6] S. Joshi and K. B. S. Manosha and M. Jokinen and T. Hänninen and Pekka Pirinen and H. Posti and M. Latva-aho, "ESC sensor nodes placement and location from moving incumbent protection in CBRS," *Proceedings of WInnComm 2016*, Mar. 2016.
[7] E. Drocella, J. Richards, R. Sole, F. Najmy, A. Lundy, and P. McKenna, "3.5 GHz exclusion zone analyses and methodology," National Telecommunications and Information Administration, Technical Report TR 15-517, Mar. 2016. [Online]. Available: http://www.its.bldrdoc.gov/publications/2805.aspx
[8] "Mathematical models for radiodetermination radar systems antenna patterns for use in interference analyses," International Telecommunication Union, Recommendation ITU-R M.1851, Jun. 2009. [Online]. Available: https://www.itu.int/rec/R-REC-M.1851-0-200906-I/en
[9] 2011 National Land Cover Database, 2011. [Online]. Available: http://viewer.nationalmap.gov/basic/
[10] 2010 National Census Tracts Gazetteer, 2010. [Online]. Available: http://www.census.gov/geo/maps-data/data/gazetteer2010.html
[11] 2010 TIGER/Line Shapefiles: Census Tracts, 2010. [Online]. Available: http://www.census.gov/cgi-bin/geo/shapefiles/index.php?year=2010&layergroup=Census+Tracts
[12] Commuter Adjusted Daytime Population: 2006-2010. [Online]. Available: http://www.census.gov/hhes/commuting/data/daytimepop.html
[13] "USGS 1-degree native format – United States." [Online]. Available: http://www.webgis.com/terr_us1deg.html
[14] "Re-sampled terrain data." [Online]. Available: http://www.its.bldrdoc.gov/resources/radio-propagation-software/resampled-terrain-data/re-sampled-terrain-data.aspx
[15] Global Land 1-km Base Elevation (GLOBE) Terrain Data. [Online]. Available: http://www.ngdc.noaa.gov/mgg/topo/globeget.html

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

An Analytical Model for Inference Attacks on the Incumbent's Frequency in Spectrum Sharing

Azza Ben Mosbah*†‡, Timothy A. Hall†, Michael Souryal†, Hossam Afifi‡
†National Institute of Standards and Technology, Gaithersburg, Maryland, USA
{azza.benmosbah, tim.hall, michael.souryal}@nist.gov
‡Télécom SudParis, Évry, France
hossam.afifi@telecom-sudparis.eu

***Abstract*—In spectrum sharing, incumbents with sensitive parameters require full protection of their operations. The incumbent's protection includes the protection of its privacy (e.g., operational frequency) against inference attacks carried out by malicious authorized secondary users. In this paper, we develop an analytical model to analyze the vulnerability of the incumbent's frequency to inference attacks and validate it by simulation. Specifically, we study random and ordered channel assignment schemes and compare results for both schemes.**

***Index Terms*—3.5 GHz band, analytical evaluation, incumbent vulnerability, inference attack, privacy, spectrum sharing.**

I. Introduction

The 3550 MHz to 3700 MHz band in the United States has a three tier access model managed by a spectrum access system (SAS) with the assistance of an environmental sensing capability (ESC). Tier 1 users includes authorized Federal users. Tier 2 and tier 3 users are collectively referred to here as secondary users (SUs). The ESC determines channel availability while the SAS manages access of SUs to the spectrum. Introducing these SUs into the band affects the privacy of the Federal incumbent. Specifically, a malicious SU may gain authorized access to the spectrum and carry out attacks to infer sensitive information about the incumbent by just fusing information given by the SAS. Such an attack is referred to as an *inference attack*.

In a previous work [1], we used simulations to model inference attacks and demonstrated that the privacy of the incumbent can be improved by adjusting the system parameters (i.e., inherent obfuscation). In this paper, we mathematically model attacks that attempt to infer the incumbent's operational frequency through repeated requests for spectrum and validate our analytical model with simulations.

II. System Model

A. Network Model

The SAS manages n channels within a given area. Only l channels are available for use by SUs. We model the SU activity using an $M/M/l/l$ queue [1]. When a SU requests spectrum resources, the SAS replies with an available channel. If no channel is available, the request is denied.

*This paper was written in partial fulfillment of Mrs. Ben Mosbah's doctoral program at Télécom SudParis (France) while working as a guest researcher at NIST.

B. Attack Model

The attacker is a legitimate SU sending queries requesting access to the spectrum. Its initial knowledge is a list of all channels in the band. Once the SAS returns an available channel, the attacker knows that the given channel is not used by the incumbent. Hence, the attacker updates its knowledge by removing it from the list of potential incumbent channels. Let X be the random variable representing the number of queries needed to discover all channels available to secondaries and, hence, the channels used by the incumbents.

III. Analytical Evaluation

In this Section, we show how to compute the expected number of queries needed to infer the incumbent's channel $E[X]$ for two SAS channel assignment schemes: one where the SAS assigns an idle channel at random, and the other where the SAS assigns the lowest-numbered idle channel [1]. In order to calculate this, we use the $M/M/l/l$ queue model and assume the system is in equilibrium.

A. Analysis of the Random Channel Assignment

In the case of the random assignment scheme, the SAS returns a channel at random in reply to an attacker's request. The blocking probability P_B is the probability that all channels are busy at the time of a request and is calculated as follows

$$P_B = \frac{\rho^l}{l!}\left(\sum_{k=0}^{l}\frac{\rho^k}{k!}\right)^{-1}, \tag{1}$$

where $\rho = \lambda/\mu$ is the system load, λ is the aggregate arrival rate, and $1/\mu$ is the individual service time.

In the above system, when the attacker makes a request, the SAS will either return one of the available idle channels or, if all the channels are busy, will respond saying no channel is available. In this case we can express $E[X]$ as

$$E[X] = E[X_b] + E[X_r]\,, \tag{2}$$

where X_b is the number of queries made when all channels were busy, i.e., the request was blocked, and X_r is the number of queries for which an available idle channel was returned by the SAS. We know that

$$E[X_b] = P_B E[X]\,. \tag{3}$$

Afifi, Hossam; Ben Mosbah, Azza; Hall, Timothy; Souryal, Michael.
"An Analytical Model for Inference Attacks on the Incumbent's Frequency in Spectrum Sharing."
Paper presented at IEEE Dynamic Spectrum Access Networks (DySPAN) Conference, Baltimore, MD. March 6, 2017 - March 9, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

All that remains is to calculate $E[X_r]$. When $1 \leq k \leq l$ channels are idle, then each channel is idle with probability k/l, and if idle will be returned by the SAS with probability $1/k$. Thus each channel has probability $1/l$ of being returned. $E[X_r]$ can then be calculated using the solution to the coupon collector's problem with equal probabilities.

$$E[X_r] = l \sum_{k=1}^{l} \frac{1}{k} = lH_l \,, \tag{4}$$

where H_l is the l^{th} harmonic number [2]. Thus, we have

$$E[X] = \frac{lH_l}{1 - P_B} \,. \tag{5}$$

B. Analysis of the Ordered Channel Assignment

In the case of the ordered assignment scheme, channels are assigned to incoming requests with unequal probabilities. We need to find the probability p_j that the SAS will return channel j in reply to an attacker's request. This is equivalent to the probability that channel j is the lowest available idle channel at the time of an attacker's request.

Let B_j be the probability that an arriving request finds the first j channels busy. The conditional probability that an arriving request finding the first $j-1$ channels busy also finds channel j busy is B_j/B_{j-1}. If $\gamma_j(z)$ is the Laplace-Stieltjes transform of the distribution function of elapsed time between successive times when an arriving request finds the first $j-1$ channels busy and is assigned channel j [3], then

$$\gamma_j(\mu) = \frac{B_j}{B_{j-1}} \,, \tag{6}$$

where $B_0 = 1$ and $\gamma_j(z)$ is defined by the recurrence relation.

$$\gamma_{j+1}(z) = \frac{\gamma_j(z+\mu)}{1 - \gamma_j(z) + \gamma_j(z+\mu)}, \quad j = 1, 2, \ldots \tag{7}$$

$$\gamma_1(z) = \tfrac{\lambda}{\lambda+z} \,.$$

Note that it follows from equation (6) and $B_0 = 1$ that

$$B_j = \gamma_1(\mu) \cdots \gamma_j(\mu), \; j = 1, 2, \ldots \tag{8}$$

Using the above, we calculate the probabilities p_j. Let I_j be a random variable representing the state of channel j, where 1 means the channel is busy and 0 means that it is idle. Then,

$$\begin{aligned} p_j &= Pr\{I_1 = 1, ..., I_{j-1} = 1, I_j = 0\} \\ &= (1 - B_j/B_{j-1})\, B_{j-1} \\ &= B_{j-1} - B_j \,. \end{aligned} \tag{9}$$

Note that $P_B = B_l$ and $P_B + \sum_{j=0}^{l} p_j = 1$.

We can find $E[X]$ by using the p_j as calculated above in the solution to the coupon collector's problem for unequal probabilities in [2]

$$E[X] = \sum_{i=1}^{l} \frac{1}{p_i} - \sum_{i<j} \frac{1}{p_i + p_j} + \sum_{i<j<k} \frac{1}{p_i + p_j + p_k} - \cdots \tag{10}$$

$$\cdots + (-1)^{l+1} \frac{1}{p_1 + \cdots + p_l} \,.$$

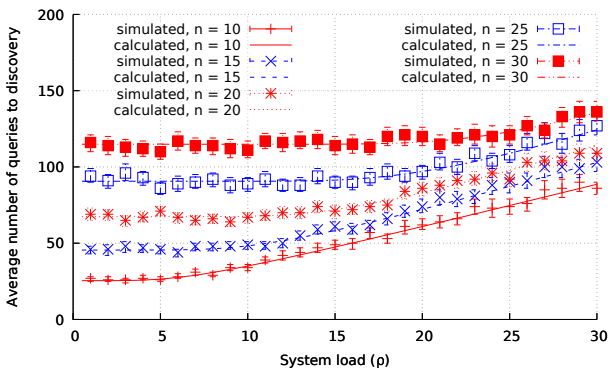


Fig. 1. Analytical results vs. simulation results for the random scheme

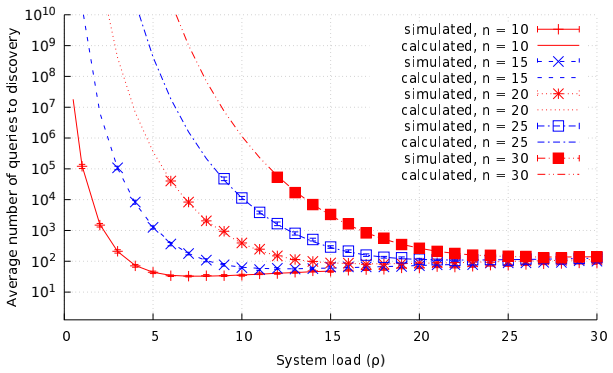


Fig. 2. Analytical results vs. simulation results for the ordered scheme

IV. Analytical Results

Our system includes one incumbent, one SAS and one attacker within the same area. The incumbent is operating on one channel. So, $n-1$ channels are available for use by SUs. We compute $E[X]$ analytically and by simulation with a confidence interval of 95 % for different values of n and ρ. In Fig. 1, where channels are randomly assigned, we note that the simulation results are within the confidence interval of the analytical model results for all values of ρ. In Fig. 2, where channels are assigned order-wise, simulation results also match the analytical model results.

V. Conclusions

We have proposed an analytical model to calculate the average number of queries needed to infer the operational channel of the incumbent. This model provides insight into the limit that the SAS may set on the query rate in order to minimize the risk of inference to the incumbent.

References

[1] A. Ben Mosbah, T. A. Hall, M. Souryal, and H. Afifi, "Analysis of the vulnerability of the incumbent frequency to inference attacks in spectrum sharing," in *IEEE Consumer Communications and Networking Conference (CCNC'17)*, Las Vegas, NV, Jan. 2017.

[2] M. Ferrante and M. Saltalamacchia, "The coupon collector's problem," *Materials matemàtics*, pp. 1–35, 2014.

[3] R. B. Cooper, "Queues with ordered servers that work at different rates," *Opsearch*, vol. 13, no. 2, pp. 69–78, 1976.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Implementation and Validation of an LTE D2D Model for ns-3

Richard Rouil, Fernando J. Cintrón, Aziza Ben Mosbah, and Samantha Gamboa
National Institute of Standards and Technology
Gaithersburg, MD, USA 20899
{richard.rouil,fernando.cintron,aziza.benmosbah,samantha.gamboa}@nist.gov

ABSTRACT

The ability to perform device-to-device (D2D) communication in Long Term Evolution (LTE)-based cellular networks became possible with the introduction of Proximity Services (ProSe) functionalities in the 3rd Generation Partnership Program (3GPP) specifications. In this paper, we provide a description of the ProSe implementation that extends the LTE model already available in ns-3. Our model contains key features defined in LTE Release 12 and further enhanced in LTE Release 13 related to synchronization, discovery, and communication. We also provide validation of each feature by comparing simulation results with analytical models developed as part of our work on D2D communication.

CCS CONCEPTS

• **Networks** → **Network performance evaluation**; **Network simulations**; **Mobile networks**;

KEYWORDS

3GPP, Long Term Evolution, Device-to-Device Communication, Network Modeling, ns-3

ACM Reference format:
Richard Rouil, Fernando J. Cintrón, Aziza Ben Mosbah, and Samantha Gamboa. 2017. Implementation and Validation of an LTE D2D Model for ns-3. In *Proceedings of the 2017 Workshop on ns-3, Porto, Portugal, June 2017 (WNS3 2017),* 8 pages.
DOI: http://dx.doi.org/10.1145/3067665.3067668

1 INTRODUCTION

Direct communication between user devices is prominent in unlicensed-band technologies such as Wi-Fi and Bluetooth. Multiple network simulation platforms, including ns-3, have implementations to support the simulation and performance evaluation of such networks.

Recently, the 3rd Generation Partnership Program (3GPP) introduced Proximity Services (ProSe) [7] into Long Term Evolution (LTE), enabling direct communication between nearby User Equipment (UEs). ProSe allows for operation in both licensed (LTE uplink spectrum) and unlicensed bands. It also supports autonomous operations for out-of-coverage public safety users. The UEs transmit and receive information without going through the evolved Node B (eNodeB) in order to synchronize, discover, and communicate with each other. Various applications and services will be enhanced by the use of device-to-device (D2D) communication. Commercially, it will be used for advertising, social networking, and gaming [1] . For public safety, D2D features will help first responders overcome service degradation due to limited resources and network failures [2]. Furthermore, D2D communication enables network operators to offload certain traffic from eNodeBs and to mitigate network congestion [12].

Analytical studies have shown that LTE D2D can provide lower delays and more energy savings compared to other short-range communication technologies such as Wi-Fi [13]. D2D communication can achieve better performance in terms of capacity, throughput, power efficiency, and spectral utilization compared to the LTE infrastructure operations [2] [18] [15]. Moreover, D2D can extend coverage when an in-coverage UE acts as a relay for other out-of-coverage UEs [1]. Given its novelty and relevance, D2D communication has been recognized as one of the key components for the fifth generation (5G) mobile networks, attracting interest from both researchers and manufacturers [8].

Currently, ns-3 offers an implementation of the LTE network, which was developed by Piro et al. [14]. We extended that implementation to support LTE D2D. In [16], we introduced our model including preliminary results on D2D communication. In this paper we provide a description of the implemented functionalities including synchronization and discovery, made available on-line[1]. Furthermore, we present validation results against mathematical models for D2D discovery and communication modules.

The rest of this paper is structured as follows. In Section 2, we provide a background on LTE D2D, outlining each of its functionalities (direct communication, direct discovery, and synchronization). In Section 3, we describe the modifications made to the ns-3 LTE module to support D2D capabilities. Section 4 presents the evaluation and validation of the model. Finally, Section 5 concludes the paper.

2 BACKGROUND

In order to support LTE D2D ProSe, 3GPP defined the PC5 interface, a new direct link between UEs called "Sidelink" at the access stratum layers. ProSe-enabled UEs can use Sidelink to exchange information when they are in close proximity. Three LTE D2D functionalities are defined under ProSe: direct communication, direct discovery, and synchronization. The direct communication functionality allows the UEs to establish a communication link between them without the need of routing the data via the eNodeBs. The direct discovery functionality allows to advertise and detect useful information provided by the UEs in proximity without the need of establishing a communication link. Finally, the synchronization functionality provides the mechanisms needed by the UEs in proximity to agree

This paper is authored by an employee(s) of the United States Government and is in the public domain. Non-exclusive copying or redistribution is allowed, provided that the article citation is given and the authors and agency are clearly identified as its source.
WNS3 2017, Porto, Portugal
ACM 978-1-4503-5219-2/17/06
DOI: http://dx.doi.org/10.1145/3067665.3067668

[1]National Institute of Standards and Technology (NIST) GitHub: https://github.com/usnistgov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

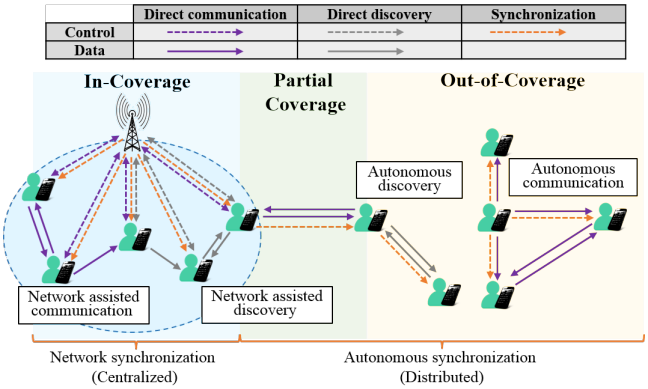


Figure 1: Overview of the LTE D2D functionalities and scenarios of operation.

on common system information and to be able to decode Sidelink transmissions.

To control the access to the D2D communication and discovery functionalities, a ProSe Function was added to the LTE core network. It is also responsible for allocating and storing discovery application identifiers (ProSe Application Code), and the processing and handling of UEs request through a new PC3 interface.

The LTE D2D functionalities can operate regardless of the network status of the UEs. Thus, three scenarios were identified by 3GPP: in-coverage, partial coverage, and out-of-coverage, as illustrated in Figure 1. When the UEs are in-coverage, the functionalities are network assisted, i.e., the UEs use the configuration and control information provided dynamically by the network, as well as preconfigured parameters. When the UEs are out-of-coverage, they rely on preconfigured parameters, enabling autonomous operations. Partial coverage, is a hybrid between the other two scenarios, in where UEs within network coverage can provide system information to out-of-coverage UEs. The following sections describe each of the LTE D2D functionalities.

2.1 Direct Communication

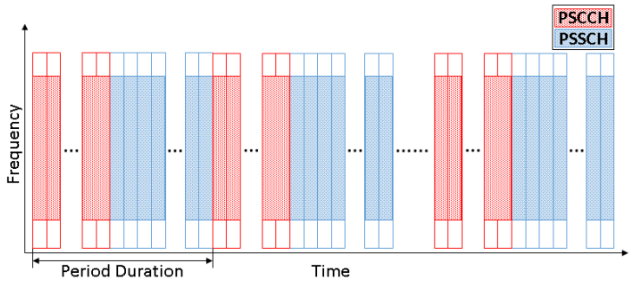


Figure 2: Sidelink communication period.

D2D communication over Sidelink is performed over periodically repeating periods in the time domain [5]. Each Sidelink period is composed of two channels spaced out in time, the Physical Sidelink Control Channel (PSCCH) and the Physical Sidelink Shared Channel (PSSCH), as depicted in Figure 2. Each channel is defined by a resource pool, i.e., a combination of certain Resource Blocks (RBs) in the frequency domain, and certain subframes in the time domain [5]. A detailed resource pool analysis for the PSCCH and its resource scheduling procedures can be found in [11].

The PSCCH is used by ProSe-enabled UEs to send a Sidelink control information (SCI) message, to indicate to who this message is addressed, how and where the data will be transmitted, i.e., the group destination ID, the modulation and coding scheme (MCS), and the PSSCH resource assignment in time and frequency, among other parameters. Each UE can be associated with one or more group IDs, and must scan through the control channel time duration to detect if another UE is going to transmit something addressed to their group. Upon successful reception of a SCI message, pertaining UEs can then proceed to tune to the corresponding resources in the PSSCH.

Transmissions in the PSSCH follow a Time Resource Pattern (TRP), which is a subframe indication bitmap of a fixed length N_{TRP} (e.g., 8 subframes) repeated through the length of PSSCH, to identify which subframes are used by a transmitting UE. Each TRP is identified by an index I_{TRP} corresponding to the predefined subframe indication bitmap established in [5]. In order to mitigate throughput degradation due to medium interference, every transmission on the PSSCH is performed with four (4) hybrid automatic repeat request (HARQ) processes without feedback. Hence, every transport block transmission on the PSSCH requires 4 subframes to be carried.

LTE Release 12 introduced two resource allocation modes, Mode 1 and Mode 2, for D2D communication. The eNodeB configures in-coverage UEs to operate on either mode, however, out-of-coverage UEs can operate only in Mode 2. In Mode 1, D2D communications are assisted by the eNodeB, i.e., resource scheduling is performed dynamically by the eNodeB. In Mode 2, UEs manage resource scheduling autonomously relying on preconfigured settings, and both, PSCCH and PSSCH, resources are selected at random from their respective resource pools.

2.2 Direct Discovery

D2D discovery, as stated in 3GPP, is a functionality that allows the detection of services and applications (e.g., gaming, social networking, advertising, etc.) offered by other UEs in close proximity [1]. It is carried independently from direct communication, as one is not required to precede the other.

D2D discovery allows discovery-enabled UEs to directly identify other neighboring discovery-enabled UEs. It can be either open or restricted depending on whether the UE needs a permission from the other discovered-to-be UE. Moreover, two models of discovery have been defined. Model A is based on an open announcement procedure where UEs broadcast information, while Model B is a request/response process used when a UE wants to ask for a certain information.

Initially, in Release 12, D2D Discovery was only supported for in-coverage scenarios for both public safety and commercial usages. However, in Release 13, 3GPP extended discovery to support out-of-coverage for public safety.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Before starting the discovery process, UEs should go through a Service Authorization and Provisioning procedure. The UE initiates the request, and the ProSe Function determines whether the UE is authorized to use ProSe direct discovery announcing (sending discovery messages), ProSe direct discovery monitoring (receiving discovery messages), or both. In the out-of-coverage case, such information is preconfigured and stored in advance in the device. Once authorized, the UEs can exchange discovery messages, called announcements [4].

The Medium Access Control (MAC) layer uses the Sidelink Discovery Channel (SL-DCH) to map the discovery message to the Physical Sidelink Discovery Channel (PSDCH). The most important component of the discovery message is the ProSe Application Code. It is allocated per announcing UE and application [3]. Another significant part related to direct discovery is the System Information Block (SIB) 19 which is transmitted by the eNodeB and provides the information about the radio resource pool where a device is allowed to announce and to monitor discovery messages [6]. ProSe enabled UEs must rely on preconfigured system information upon the absence of SIB 19.

The discovery resource pool determines the discovery period that could be up to 1024 radio frames (10.24 seconds) long. A discovery offset is defined to delay the discovery process with respect to the beginning of the period. The resource pool includes a bitmap that indicates which subframes (noted SF) could be used for discovery, and the number of times this bitmap must be reused within the discovery period. In Figure 3, SF_j represents one subframe (i.e. time slot) allocated to discovery and N_t defines the total number of such subframes.

A resource configuration for the frequency domain is also provided. It defines the total number of RBs dedicated to discovery (N_f) and the associated start and end numbers (RB_{start} and RB_{end} respectively). This allows the organization of the discovery bandwidth in clusters as shown in Figure 3 and computed in [5]. We also note that the discovery message can be retransmitted several times, with the number of retransmissions being configurable between 0 and 3 [6].

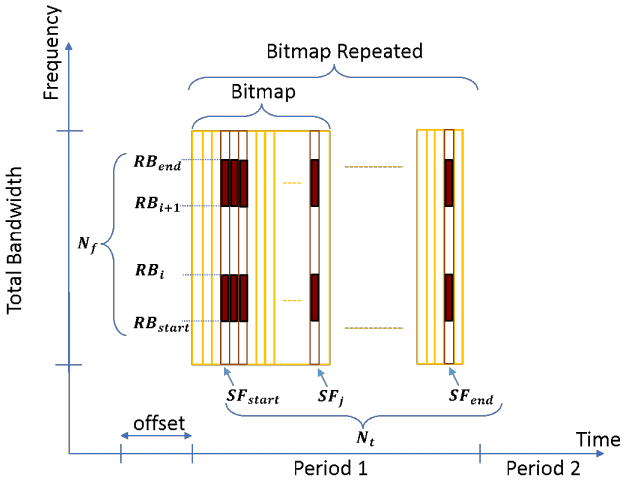


Figure 3: Resource pool configuration for discovery.

Two discovery resource allocation types are defined, Type 1 and Type 2B. In Type 1, (i.e., "UE-Selected"), UEs select independently and arbitrarily the discovery resources to transmit discovery messages. Type 2B (i.e., "Scheduled") represents a UE-dedicated resource allocation provided by the eNodeB [2].

2.3 Synchronization

In order to establish effective direct communication and discovery, UEs need to be aligned in time and frequency, and they need to agree on the same system information used in the communication procedures (e.g., bandwidth, subframe indication, etc.). Thus, two UEs attempting to communicate need to follow the same Synchronization Reference (SyncRef). If the UEs are in-coverage, their SyncRef is provided by the eNodeB and the synchronization configuration can be found inside the SIB18 and SIB19 messages for communication and discovery respectively. When the UEs are out-of-coverage, preconfigured parameters are used to initiate the synchronization process and to agree on a common SyncRef, giving priority to those originating from in-coverage UEs (e.g., partial coverage scenario), if available.

The Sidelink synchronization information transmission procedure defines when a UE should be a SyncRef and announce the synchronization information [6]. When in-coverage, the UE becomes a SyncRef if the eNodeB explicitly instructs it, or if the perceived eNodeB signal strength is below a given threshold and the UE is transmitting in the Sidelink. An out-of-coverage UE becomes a SyncRef if it is transmitting in the Sidelink and either it does not have a selected SyncRef, or the signal strength of the selected SyncRef is below a given threshold. Otherwise, the UE will cease to be a SyncRef or will not become one.

When the UE becomes a SyncRef, it periodically transmits Sidelink Synchronization Signals (SLSS) for announcing its synchronization information. An SLSS is transmitted in one subframe in the time domain and uses the central 6 RBs in the frequency domain. The periodicity of the SLSS is 40 ms, and the exact time slot is indicated by a relative subframe offset present in the synchronization configuration.

An SLSS is composed of four elements: The Primary Sidelink Synchronization Signal (PSSS), the Secondary Sidelink Synchronization Signal (SSSS), the Demodulation Reference Signals (DMRS), and the Physical Sidelink Broadcast Channel (PSBCH). The PSSS and SSSS are used for time and frequency reference; together they encode the SLSS identifier (SLSSID), which identifies the SyncRef. There is a subset of SLSSIDs reserved for identifying SyncRefs in-coverage (configured by the network) and another subset reserved for out-of-coverage use. The PSBCH carries the MasterInformationBlock-SL (MIB-SL), which contains the system level information needed for the configuration of the synchronizing UE. The DMRSs are used as a reference for channel estimation, demodulation of the PSBCH, and measurement of Sidelink Reference Signal Received Power (S-RSRP) in the receiving UE. The S-RSRP is the indicator of the SyncRef signal strength.

The UEs search for available SyncRefs, measure the S-RSRP of the detected ones if any, and synchronize to the most adequate one according to the Sidelink synchronization reference procedure. A SyncRef is considered detected if the UE has obtained its SyncRef

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

SLSSID and has decoded its MIB-SL. After the measurement and filtering process, a SyncRef is considered valid if its S-RSRP exceeds a predefined minimum required threshold by a given hysteresis value. The UE ranks the valid detected SyncRefs by their priority group and their S-RSRP values. The priority group is determined by their SLSSID and their network condition (in-coverage or out-of-coverage, indicated inside the MIB-SL). In-coverage SyncRefs have the highest priority, followed by out-of-coverage SyncRefs with SLSSID from in-coverage (i.e., the SyncRef is using in-coverage synchronization information even if it is out-of-coverage), and the lowest priority is for pure out-of-coverage SyncRefs.

When the UE does not have a valid selected SyncRef, it will synchronize to the SyncRef with highest priority group and strongest S-RSRP. If the UE already has a valid selected SyncRef, the UE compares it with the candidate SyncRef with the strongest S-RSRP. The UE selects the candidate SyncRef if it belongs to a higher priority group than the currently selected SyncRef, or if it belongs to the same priority group, but its S-RSRP exceeds the one of the selected SyncRef by a given threshold. Otherwise, the UE keeps the selected SyncRef. In out-of-coverage scenarios, the convergence to a unique SyncRef within a group of UEs is challenging, as it is a distributed process influenced by different parameters. The synchronization framework in our model was used in previous work to evaluate the out-of-coverage synchronization performance [9].

3 IMPLEMENTATION

In this section we describe the additions and modifications made to the ns-3 LTE module (version 3.22) to support the D2D functionalities. An overview of the changes is shown in Table 1 and detailed descriptions for each module are provided in the following subsections.

3.1 Non Access Stratum (NAS)

The functionalities of the NAS layer include Evolved Packet System (EPS) mobility management and session management by exchanging messages between the UE and the core network. Since ns-3 supports only one core, the mobility management functions are not implemented in the EpcUeNas class. Existing functionalities include the activation of EPS bearers, filtering of UL data packets, and the transmission/reception of data packets. A bearer is associated with one or several Traffic Flow Templates (TFTs) used to define the rules mapping IP packets to the right bearer based on IP addresses, ports, and type of service parameters.

For D2D communication, there is no EPS bearer setup. Thus, the model includes new functions to activate Sidelink bearers. A new type of TFT, called SlTft, maps IP packets to the Sidelink bearers based only on the IP destination address of the packets. The Send function has also been modified to allow the transmission of packets even when the state of the NAS layer is OFF to support out-of-coverage scenarios.

3.2 Radio Resource Control (RRC) Protocol

The RRC layer provides signaling between the eNodeB and the UE to perform attachment and setup radio bearers. Regarding D2D, it also contains the resource pools' configurations used for communication, discovery, and synchronization. Modifications were made to both the eNodeB side, via the LteEnbRrc class, and the UE side, via the LteUeRrc class.

On the eNodeB side, SIBs 18 and 19 were added to broadcast Sidelink resource pool configurations for communication and discovery respectively. The configuration of the resource pools is done using functions added to the LteHelper. The eNodeB is also now capable of processing SidelinkUeInformation messages sent by the UE. This type of message contains information associated with the demand and management of resources for communication and discovery, such as the identity of the destination(s) and the number of resources requested (for the "Scheduled" mode). The response is sent by the eNodeB using the RrcConnectionReconfiguration message.

Regarding the UE RRC layer, new functions were added to support the creation of Sidelink bearers and the modification (creation/removal) of discovery applications, regardless of the UE network state (in-coverage or out-of-coverage). UEs filter the received discovery messages based on the applications that they are interested in monitoring.

The extended model is also capable of processing the new SIB18 and SIB19 messages defined in the LteRrcSap class, and the dedicated Sidelink configuration received in RrcConnectionReconfiguration messages.

The synchronization protocol logic described in Section 2.3 was mainly implemented in the LteUeRrc class. The most relevant functionalities of the developed model are: the activation and deactivation of the SLSS transmission according to the Sidelink synchronization information transmission procedure, the configuration of the SLSS to be transmitted by the physical layer protocol, the reception of the SyncRefs measurement report, the selection of the SyncRef according to the Sidelink synchronization reference procedure, and the instruction to change SyncRef to the other layers.

3.3 Packet Data Convergence Protocol (PDCP)

The PDCP layer available in ns-3 already supports the Unacknowledged Mode (UM) transmission used by the Sidelink bearers. However, a logical channel within a UE can no longer be identified uniquely by its logical channel identifier (LCID). With D2D communication, UEs create new logical channels for each destination to which they are transmitting (i.e. Layer 2 group ID), assigning LCIDs independently. It is possible that multiple UEs select the same LCID for the same group so receiving UEs must identify the remote UE for which it receives packets. Therefore, the identifiers for the logical channels have been extended to include the source Layer 2 ID and destination Layer 2 ID that identify the transmitter UE and the group to which the packets must be delivered.

3.4 Radio Link Control (RLC) Protocol

The modifications made to the RLC layer, namely class LteRlc, are identical to the PDCP layer, where new identifiers were added to support Sidelink bearer identification.

3.5 Medium Access Control (MAC) Protocol

The MAC protocol is responsible for allocating radio resources. To support the scheduled mode where the eNodeB allocates resources

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Table 1: Overview of the main changes and extensions introduced with the LTE D2D model implementation

ns-3 class	Direct communication	Direct Discovery	Synchronization
EpcUeNas	- Management of sidelink bearers - Transmission of packets when out-of-coverage		
LteEnbRrc	- Transmission of *SIB18* message	-Transmission of *SIB19* message	
	- Transmission of *RrcConnectionReconfigurationmessage* - Processing of *SidelinkUeInformation* message		
LteUeRrc	- Creation of sidelink bearers - Reception and processing of *SIB18* message - Transmission of *SidelinkUeInformation* message	- Creation and removal of discovery applications - Reception and processing of *SIB19* message - Filtering of discovery messages - Tracing of discovered applications	- UE synchronization status tracking - Execution of the sidelink synchronization information transmission procedure - Execution of the SyncRef selection procedure
	- Reception and processing of *RrcConnectionReconfiguration* message		
LtePdcp	- Extension of the logical channel identifiers		
LteRlc	- Extension of the bearer identifiers		
LteEnbMac	- Reception and processing of *BSR* messages (Mode 1) - Generation and transmission of traffic scheduling allocation (Mode 1)		
LteUeMac	- Generation of traffic scheduling allocation (Mode 2) - Coordination of the sidelink transmissions (Mode 1 and 2)	- Creation and scheduling of discovery messages	- Timing information updating upon synchronization
LteUePhy	- Reception of transmissions in the uplink channel - Implementation of half-duplex mode		
	- Monitoring, reception, and transmission of *SCI* messages - Coordination of the PSSCH reception according to announcement in the *SCI* messages	- Monitoring, reception, and transmission of discovery messages	- Monitoring, reception, and transmission of SLSSs - Control S-RSRP measurement and report process - Timing information updating upon synchronization
LteSpectrumPhy	- Advanced interference calculation for sidelink transmissions		
LteHarqPhy	- Integration of sidelink physical layer error models		

for the UEs transmitting on the Sidelink, the implementation modifies both the eNodeB (LteEnbMac) and the UE (LteUeMac) classes. The changes include the handling of Sidelink Buffer Status Requests (BSRs) that indicate how much D2D traffic needs to be transmitted, and the schedulers to handle the Sidelink resource allocations (i.e. Mode 1). The interface between the RRC and MAC layers has also been modified to allow the RRC layer to manage the resource pools that can be used to schedule resources. A sample scheduler based on the existing Round Robin implementation is provided in class RrSlFfMacScheduler.

To perform D2D functionalities out-of-coverage or when in-coverage but the allocation mode is UE selected (i.e. Mode 2), UEs have to handle their own resource scheduling decisions. The current implementation adds attributes to the LteUeMac to configure the number of resource blocks, subframes, and MCS to be used in each Sidelink period where a Sidelink transmission occurs. The UE MAC also receives notification from the RRC layer about changes in SyncRef. When this occurs, the MAC updates the timing information (frame and subframe references) accordingly.

The discovery messages are created in the LteUeMac using information (i.e., ProSe Application Code) configured by the LteHelper. Discovery resources are assigned via the "UE-Selected" mode based on the resource pool defined in the scenario. "Scheduled" mode is not implemented yet.

3.6 Physical Layer (PHY) Protocol

The UE physical layer has been significantly modified to support the Sidelink features. To address the need for UEs to receive transmission in the uplink channel, since the Sidelink is using the uplink frequency, the LteUePhy includes an additional instance of LteSpectrumPhy that connects to the uplink channel. Since it is difficult for a UE to send and receive at the same time on the same frequency due to self-interference, the model uses half-duplex mode on the Sidelink by default. This constraint can be removed by changing the value of the FullDuplexEnabled attribute. Information about the Sidelink and discovery resource pools have also been added to the physical layer in order to compute the boundaries of both the communication and discovery periods, to appropriately monitor the resources associated with the PSCCH, PSSCH and PSDCH, as well to perform Layer 1 filtering. The physical layer processes the SCI messages received on the PSCCH to determine when data may arrive on the PSSCH and provide the information to the LteSpectrumPhy class. Discovery and synchronization have been implemented as broadcast processes where monitoring UEs receive all messages sent of the relevant types, and pass them to the upper layers, that will perform the appropriate filtering.

The LteUePhy class was extended to handle the transmission and reception of SLSSs, the S-RSRP measurement and report, and the change of timing due to SyncRef (re)selection. In our model, the whole SLSS is represented inside the MIB-SL, which is modeled as a control message comprising all the MIB-SL fields defined in [6] plus two metadata fields: the SLSSID of the SyncRef, and a reception timestamp to emulate the time acquisition that real systems do upon detection of the PSSS and SSSS. Regarding the calculation of the S-RSRP, we use the same approach to the one already implemented in ns-3 for the downlink RSRP measurement. In case that a change of SyncRef is instructed by the RRC, the PHY updates the timing information (frame and subframe references) accordingly.

The introduction of additional physical channels for Sidelink lead to the development of physical layer error models. The LTE toolbox in Matlab was extended and the results were integrated in ns-3. A complete description of the methodology and resulting models

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

are available in [17]. To handle the new error models, the LteHarqPhy class was also modified to store the Signal-to-Interference Noise Ratio (SINR) value of each transmission that is used as part of the soft combining process during retransmissions. Another key enhancement is the handling of collisions/interference. The LTE interference model available in ns-3 computes the SINR for each incoming transmission. Its design assumes that there is no frequency overlap between the transmissions coming from/to an eNodeB within a subframe because the eNodeB does all the scheduling. Transmissions from other eNodeB/UEs are simply considered interference. With ProSe, out-of-coverage UEs or those in "UE-selected" mode can select the same (or overlapping) resources because the allocation is uncoordinated. In order to determine which packet will be successfully decoded, the new implementation keeps track of the SINR values for each Sidelink transmission.

3.7 Channel Models

To more accurately model the propagation loss that occurs in a D2D transmission, several D2D propagation models have been implemented as specified by 3GPP [2]. They capture outdoor to outdoor (3gppOutdoorPropagationLossModel), outdoor to indoor (3gppHybridPropagationLossModel), and indoor to indoor (3gppIndoorPropagationLossModel) environments. While each model can be used individually in a scenario, another propagation model, 3gppPropagationLossModel, has also been created as a wrapper that evaluates the conditions between the transmitter and receiver (i.e. are they indoor or outdoor), and calls the appropriate model, thus supporting heterogeneous deployments.

3.8 Helpers

The LteHelper class is used to facilitate the creation of scenarios involving LTE nodes. The helper class has been updated to support all the changes presented previously, mainly regarding the creation of UE devices for which the internal structure has changed. It installs the Sidelink configuration containing the D2D resource pools and the discovery applications after being defined in the scenario. It also allows to configure the initial time references (SLSSID, and frame and subframe numbers) of a group of UEs.

4 VALIDATION

4.1 Direct Communication

4.1.1 PSCCH. The selection of resources to transmit a SCI message over the PSCCH is performed randomly without any feedback or collision avoidance mechanism when operating in Mode 2. Given a PSCCH resource pool size (N_{PSCCH}), and the number of UEs (n_{UE}) contending to transmit, we can compute the likelihood of a collision to occur among UEs' resource selection as:

$$P_{Collision} = 1 - \binom{N_{PSCCH}}{n_{UE}} \times \frac{n_{UE}!}{N_{PSCCH}{}^{n_{UE}}} \quad (1)$$

The selection of a resource by two or more UEs operating in the same Sidelink period is considered as a collision. To validate our PSCCH model, two PSCCH pool configurations yielding 176 and 880 transmission resources, respectively, were simulated. For each configuration multiple scenarios were simulated with a fixed number of deployed UEs contending for D2D transmission, lasting 1000 Sidelink periods. The collision rate for PSCCH resource selection was evaluated against the theoretical values for each scenario. Figure 4 shows the average collision rate from the simulations, including 95 % confidence intervals, to lineup with the theoretical values. As expected, the probability of a PSCCH collision increases as the number UEs contending for pool resources increases.

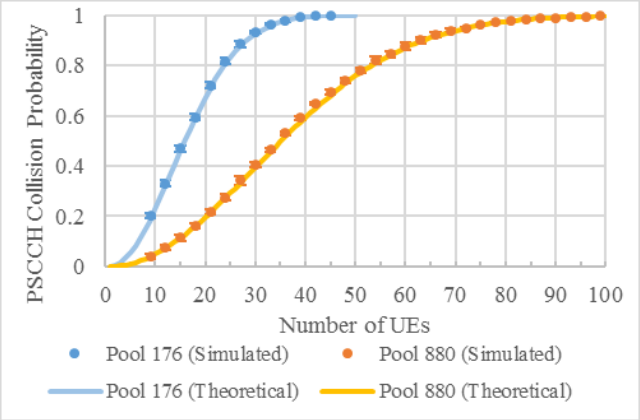


Figure 4: PSCCH resource collision validation.

4.1.2 PSSCH. We validate the implementation of the shared channel by comparing the theoretical data rates to the ones obtained in simulations. We compute the theoretical data rate, in bit/s, for a given MCS and RB allocation as follow:

$$rate = \frac{tbSize \times \left\lfloor \frac{(period - PSCCH) \times KTRP}{8 \times N_{HARQ}} \right\rfloor}{period} \quad (2)$$

Where the tbSize is the transport block size defined in [5] (in bits), period is the duration of the Sidelink (in ms), PSCCH is the size of the control channel (in ms), KTRP is the number of active subframes in the repetition pattern, and $N_{HARQ} = 4$. In the simulation scenario, we place a transmitter and a receiver such that there is no packet loss due to the channel conditions. Three configurations, shown in Table 2, are used to validate that the model takes into account the Sidelink configurations. Figure 5 shows that the data rates obtained in simulation are perfectly matching the theoretical values for all configurations.

Table 2: Simulation parameters for PSSCH validation

Configuration	SL period (ms)	PSCCH duration (ms)	KTRP	MCS
1	40	8	2	10
2	80	8	4	12
3	320	40	8	15

4.2 Direct Discovery

In this Section, we validate the D2D direct discovery's implementation in UE-Selected mode. We compare the simulation results to the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

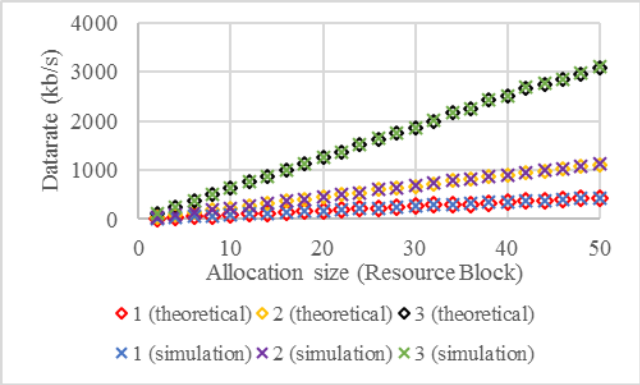


Figure 5: PSSCH data rate validation.

analytical model elaborated by Griffith et al. [10]. The mathematical study considered a group of UEs interested in both announcing their own application and monitoring everyone else's application. The exchange of discovery messages occurs using a simplistic propagation environment: all the UEs were able to detect each other and all colliding announcements (using same discovery resources) were discarded. The model also took into account the half duplex limitation where a UE could not receive any discovery message from another UE if both UEs happened to transmit announcements in the same time slot (i.e., a subframe). Under those assumptions, the time needed for a random UE to discover all other UEs in its group is computed.

In our simulations, we considered a resource pool consisting of 5 subframes and 10 pairs of resource blocks. We deployed 10 UEs, distributed randomly within an area of 100 m x 100 m. For this scenario, we performed 10 runs, with 500 trials per run. We computed the number of periods needed for one random UE to discover everyone else in the group. The average of the results collected from all the runs was used to generate the Cumulative Distribution Function (CDF) of the number of periods needed for one UE of the discovery group to discover all other UEs.

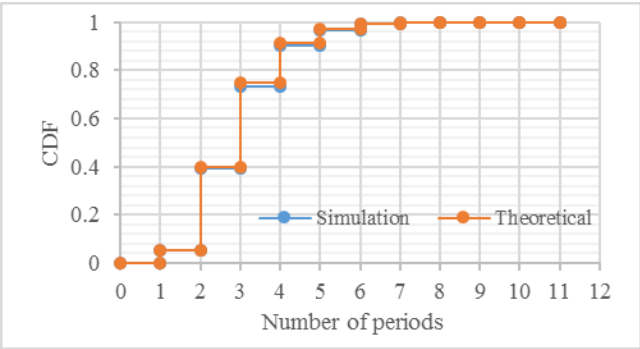


Figure 6: Number of periods needed for one random UE to discover all other UEs in the group: 50 resources and 10 UEs.

The ns-3 simulation results agree with the theoretical results as shown in Figure 6. Further resource pool configurations and group sizes had been tested in [10] and the corresponding results matched as well.

4.3 Synchronization

We tested different values of the synchronization protocol parameters in order to evaluate if the implementation follows the expected behavior. We used the outdoor uniform two ring topology described in [2] with 63 transmitter UEs, and we repeated each experiment 50 times with different random seeds. We consider all UEs out-of-coverage and unsynchronized at the beginning of the simulation. We considered two scenarios depending on the transmitter application: always-on, where the UE is sending full buffer traffic all the time; and on-off, where the transmitter is active intermittently, following the pattern for voice traffic specified in [2].

Figure 7 shows the number of SyncRef UEs in the system, along with 95 % confidence intervals, for various values of the parameter syncTxThreshOoC. This parameter is the threshold that out-of-coverage UEs use to determine if they have to become a SyncRef and transmit SLSSs. We observe that the larger the value of syncTxThreshOoC, the larger the number of UEs acting as SyncRefs. This is the expected behavior, as the range of selected SyncRef S-RSRP values causing that the UE acts a SyncRef is larger when the threshold value is larger. Given the uniform distribution of the UEs, the larger the range the more UEs detect their SyncRef within that range and act as SyncRefs.

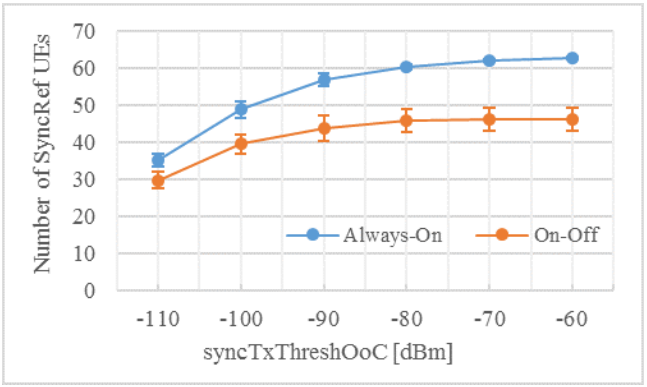


Figure 7: Number of SyncRef UEs in the scenario after 450 SyncRef selection cycles for different values of the parameter syncTxThreshOoC.

5 CONCLUSION AND FUTURE WORK

In this paper, we presented the extensions made to the ns-3 LTE implementation to support D2D synchronization, discovery, and communication as defined by ProSe. The behavior of the model has been verified using multiple scenarios, and D2D discovery and communication were validated using mathematical models. This model is being developed as part of an ongoing research in Public Safety Communications and 5G technologies. While we continue the work carried out so far, our goal in releasing the model on the NIST GitHub, is to accelerate its development, and to include new features such as relays, priority queueing, and additional transmission modes used for vehicular-to-anything (V2X).

REFERENCES

[1] 3GPP. 2013. *Technical Specification Group Services and System Aspects; Feasibility Study for Proximity Services (ProSe); TR 22.803.* Technical Report. Third Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/22803.htm

[2] 3GPP. 2015. *Technical Specification Group Radio Access Network; Study on LTE Device to Device Proximity Services; Radio Aspects; TR 36.843.* Technical Report. Third Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/36843.htm

[3] 3GPP. 2016. *Technical Specification Group Core Network and Terminals; Numbering, Addressing and Identification; TS 23.003.* Technical Report. Third Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/23003.htm

[4] 3GPP. 2016. *Technical Specification Group Core Network and Terminals; Proximity-services (ProSe) User Equipment (UE) to ProSe Function Protocol Aspects; Stage 3; TS 24.334.* Technical Report. Third Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/24334.htm

[5] 3GPP. 2016. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures; TS 36.213.* Technical Report. Third Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/36213.htm

[6] 3GPP. 2016. *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol Specification; TS 36.331.* Technical Report. Third Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/36331.htm

[7] 3GPP. 2016. *Technical Specification Group Services and System Aspects; Proximity-based Services (ProSe); Stage 2; TS 23.303.* Technical Report. Third Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/23303.htm

[8] B. Bertenyi. 2014. 3GPP System Standards Heading into the 5G era. (2014). http://www.3gpp.org/news-events/3gpp-news/1614-sa

[9] S. Gamboa, F. J. Cintrón, D. W. Griffith, and R. Rouil. 2017. Impact of Timing on the Proximity Services (ProSe) Synchronization Function. In *14th Annual IEEE Consumer Communications & Networking Conference (CCNC 2017).* Las Vegas, NV, USA.

[10] D. W. Griffith, A. Ben Mosbah, and R. Rouil. 2017. Group Discovery Time in Device-to-Device (D2D) Proximity Services (ProSe) Networks. In *IEEE Conference on Computer Communications (INFOCOM 2017).* Atlanta, GA, USA, Forthcoming.

[11] D. W. Griffith, F. J. Cintrón, and R. Rouil. 2017. Physical Sidelink Control Channel (PSCCH) in Mode 2: Performance Analysis. In *EEE International Conference on Communications (ICC 2017).* Paris, France, Forthcoming.

[12] A. Hematian, W. Yu, C. Lu, D. Griffith, and N. Golmie. 2016. A Clustering-Based Device-to-Device Communication to Support Diverse Applications. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems - RACS '16.* ACM Press, New York, New York, USA, 97–102. DOI: https://doi.org/10.1145/2987386.2987391

[13] L. Militano, M. Condoluci, G. Araniti, A. Molinaro, A. Iera, and F. H.P. Fitzek. 2014. Wi-Fi Cooperation or D2D-based Multicast Content Distribution in LTE-A: A Comparative Analysis. In *2014 IEEE International Conference on Communications Workshops (ICC).* IEEE, Sydney, Australia, 296–301. DOI: https://doi.org/10.1109/ICCW.2014.6881212

[14] G. Piro, N. Baldo, and M. Miozzo. 2012. An LTE Module for the ns-3 Network Simulator. In *3rd International Workshop on ns-3.* ACM, Barcelona, Spain. DOI: https://doi.org/10.4108/icst.simutools.2011.245571

[15] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy. 2013. Proximity-Based Data Offloading via Network Assisted Device-to-Device Communications. In *2013 IEEE 77th Vehicular Technology Conference (VTC Spring).* IEEE, Dresden, Germany, 1–5. DOI: https://doi.org/10.1109/VTCSpring.2013.6692723

[16] R. Rouil, F. J. Cintrón, A. Ben Mosbah, and S. Gamboa. 2016. A Long Term Evolution (LTE) Device-to-device Module for ns-3. In *Workshop on ns-3 (WNS3).* Seattle, WA, USA.

[17] J. Wang and R. Rouil. 2016. *BLER Performance Evaluation of LTE Device-to-Device Communications.* Technical Report. National Institute of Standards and Technology, Gaithersburg, MD. DOI: https://doi.org/10.6028/NIST.IR.8157

[18] C.-H. Yu, K. Doppler, C. B. Ribeiro, and O. Tirkkonen. 2011. Resource Sharing Optimization for Device-to-Device Communication Underlaying Cellular Networks. *IEEE Transactions on Wireless Communications* 10, 8 (aug 2011), 2752–2763. DOI: https://doi.org/10.1109/TWC.2011.060811.102120

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Information Technology Laboratory

Work of researchers at professional conferences as reported in Fiscal Year 2017

Application Creation for an Immersive Virtual Measurement and Analysis Laboratory

Wesley N. Griffin* William L. George* Terence J. Griffin* John G. Hagedorn* Marc Olano*
Steven G. Satterfield* James S. Sims* Judith E. Terrill*

Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, US

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

ABSTRACT

Content creation for realtime interactive systems is a difficult problem. In game development, content creation pipelines are a major portion of the code base and content creation is a major portion of the budget. In research environments, the choice of rendering and simulation systems is frequently driven by the need for easy to use content authoring tools. In visualization, this problem is compounded by the widely varying types of data that users desire to visualize. We present a visualization application creation framework incorporated into our visualization system that enables measurement and quantitative analysis tasks in both desktop and immersive environments on diverse input data sets.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; I.3.4 [Computer Graphics]: Graphics Utilities—Application packages

1 INTRODUCTION

A key part of laboratory science is measurement: taking measurements of data and ensuring those measurements are repeatable. Metrology is the science of measurement and has traditionally meant measurements on laboratory data that are made during the course of a physical experiment. Computational Science is the growing field of applying large-scale numerical simulation codes to develop and expand scientific understanding. With computational science, a mathematical model is simulated on a computer and, while physical experiments are used to validate the model and simulation, there is usually no corresponding physical experiment to the running simulation. This creates the issue of how to perform measurement and analysis on the simulation data.

One possible output from the measurement and analysis process are reference materials which can be used to calibrate or assess other measurement systems or processes [11]. Reference materials can be certified which include certificates attesting to the validity and traceability of the metrology process for creating the reference material. Reference materials are important for national standards organizations in helping industrial companies ensure machines are calibrated.

We have developed a virtual measurement and analysis laboratory in a 3D immersive virtual environment using a CAVE hardware configuration and a software visualization system [4, 7, 13]. One of the challenging aspects during the development and deployment of this system has been managing the widely disparate types of data that scientists want to measure and analyze. This "content creation problem" is not unique to our system. In general, content creation for realtime interactive systems is a difficult problem.

In the game development industry, content creation and "asset pipelines" are a major portion of the development team, time, and budget. The software developed for content creation often has very limited reusability. In research environments, there is often no budget for content creation (be it time or money) and thus the ability to quickly create content often forces researchers to use commercial-off-the-shelf (COTS) rendering systems that provide easy content creation tools. However, being forced to use COTS rendering systems can limit the flexibility of the realtime interactive system.

We present our High End Visualization (HEV) system which is a flexible framework for visualization application creation. Our system design provides two main abstraction points: renderer display output and run-time renderer control. Abstracting renderer display output enables our system to run a single binary on both desktop workstations and immersive environments without change to the visualization application. Abstracting run-time renderer control enables our system to provide a large set of small, flexible tools that can be composed with input data to easily develop visualization applications.

Section 2 introduces our visualization system software architecture. A key feature of our system is the use of a named pipe and text-based control commands for run-time modification of the in-memory scenegraph. This named pipe and set of control commands combine with a large set of application-generic scripts and tools to create our flexible content creation framework with fast iteration for visualization application development. Section 3 describes our content creation framework in detail. Section 4 describes a few of the visualization applications that have been developed with HEV and are currently in use.

2 SYSTEM ARCHITECTURE

Our CAVE consists of three display surfaces, 3D stereoscopic projectors with synchronized glasses, 6 DOF tracking for the user's head, a 6 DOF tracked wand for user interaction, and two channel audio. Two of the display surfaces are rear-projection screens mounted vertically at 90 degrees to each other. The third surface is a front-projection floor suitable for walking or standing. These surfaces are configured as a corner as shown in Figure 1. The corner configuration has proved very effective over the years. It is compact, minimizing floor space requirements, yet provides a comfortable space for collaborative working sessions and demonstrations.

Our visualization system is called HEV (High End Visualization) and consists of a stack of vendor supplied, open source, and in-house software pieces. Figure 2 shows how these pieces fit together. Currently we use OpenSceneGraph (OSG) for rendering and have built two layers on top of OSG: Interpreted Runtime Immersive Scenegraph (IRIS) and IRIS Development Environment for Applications (IDEA) which we describe next.

2.1 Interpreted Runtime Immersive Scenegraph

On top of OSG we have developed the Interpreted Runtime Immersive Scenegraph (IRIS) which controls the hardware elements of our CAVE. IRIS creates the rendering contexts to drive the CAVE displays and also integrates the hardware tracker data to build the user view and update the projected images. The IRIS command `iris-viewer` is the main program that initializes the CAVE and

*e-mail: {firstname.lastname}@nist.gov

George, William; Griffin, Terence; Griffin, Wesley; Hagedorn, John; Olano, Thomas; Satterfield, Steven; Sims, James; Terrill, Judith.
"Application Creation for an Immersive Virtual Measurement and Analysis Laboratory."
Paper presented at Workshop on Software Engineering and Architectures for Realtime Interactive Systems,
Greenville, SC. March 19, 2016 - March 23, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Figure 1: The CAVE corner configuration (left) has proved very effective over the years. It is compact, minimizing floor space requirements, yet provides a sufficient space for collaborative sessions and demonstrations and the wide field of view provides good immersion. Using the CAVE (right) to analyze Newtonian and Non-Newtonian flows in a pipe. See Section 4 for discussion of this specific visualization application.

performs all of the rendering. IRIS has two key features to enable rapid and flexible visualization application development: Display Object and Control Pipe.

Display Object IRIS uses dynamic shared objects loaded at runtime to configure the rendering contexts and provide the graphics windows for rendering. By separating the display devices into dynamic shared objects, IRIS can execute the same binary on both the CAVE hardware and development workstations which have standard 2D monitors. This enables visualization application developers to sit at their desks and develop applications on top of HEV that can then run in the CAVE with no changes needed. For testing and debugging, IRIS provides a simulator mode that simulates the CAVE hardware on a workstation with multiple graphics windows.

Control Pipe IRIS also listens on a named pipe for text-based control commands. These commands can modify the in-memory scenegraph thus enabling separate processes to dynamically update the rendered visualization. Since the control pipe can be written to by any other process or script, the majority of our visualization applications make heavy use of control commands. Using scripts to modify the scenegraph at run-time provides fast iteration on visualization application development and content creation.

2.2 IRIS Development Environment for Applications

On top of IRIS, HEV provides the IRIS Development Environment for Applications (IDEA). This is an in-house collection of commands, scripts, and file formats. IDEA is developed using the traditional UNIX philosophy of small tools focused on one simple task that is useful for a wide variety of applications. A nice feature of the IRIS Control Pipe is that IDEA can leverage standard software packages. IDEA uses X11 and a standard window toolkit for providing the visualization application graphical user interface (GUI). The 2D GUI windows are overlaid on top of the 3D rendering window and thus easy to manipulate in the CAVE. Figure 3 shows how the wand controls GUI menus.

As an example, the IDEA tool `irisfly` is a shell script that wraps and extends the IRIS command `iris-viewer`. `irisfly` is a general tool for displaying a wide variety of data with all of the capabilities of the CAVE. As needed, an application-specific file loaded can be created and added to `irisfly` and `irisfly` can be easily combined with other HEV tools to assemble specific visualization applications.

Another example of a widely used generic IDEA tool is the IDEA clipping controls. IRIS configures eight clipping planes in

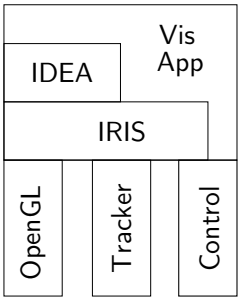


Figure 2: Software stack.

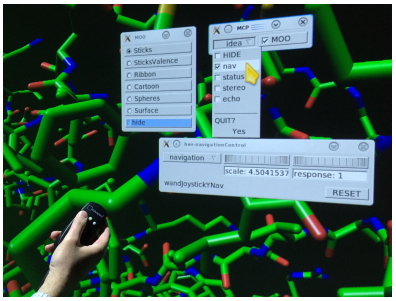


Figure 3: Wand control of GUI menus.

the scenegraph with no specific geometry that are disabled by default. The clipping control tool implements several different clipping geometries, such as a windshield, corner, or box, and dynamically controls the clipping planes using the Control Pipe.

2.3 Visualization Applications

The Visualization Applications sits atop the software stack and are frequently assembled by combining tools from IDEA and issuing commands to the Control Pipe. All current applications are based on `irisfly` and implemented without modification or recompiling the core iris-viewer code. Since separate processes can write to the Control Pipe, visualization applications can leverage additional software packages such as the R statistical analysis tool, or the D3.js information visualization tool. Section 3 discusses in more detail how IDEA and the IRIS Control Pipe combine to create a flexible content creation pipeline with fast iteration. Section 4 describes a few of the visualization applications that have been developed with HEV and are currently in use.

3 Visualization Application Creation

In HEV, a new visualization application typically starts as a shell script that processes the input data files. For input data types that have been used in the past, there is likely an IDEA tool for processing the files. If the input data has not been used with HEV before, our process is to develop any application-specific tools for processing the data and then abstract those tools into general tools for future use.

```
FUNC RgbaLut1D myTF2
    VAL_TRANSFORM log10
    RGBLUT
      1.2e-20   0.4 0.1 0.7
      3.9e-15   0.2 0.2 0.6
      6.8e-07   0.1 0.6 0.6
      5.9e-02   0.1 0.6 0.2
      1.0e+01   0.1 0.7 0.1
    END_RGBLUT
    ALPHALUT
      1.2e-20   0.0
      3.9e-15   0.4
      6.8e-07   0.9
      5.9e-02   1.0
      1.0e+01   1.0
    END_ALPHALUT
END_FUNC myTF2
```

Figure 4: Example of the transfer function description. The volume data is assumed to have a single scalar value at each voxel location.

3.1 Input Data Processing

Input data processing results in geometry files for rendering. The geometry files are frequently saved as OSG binary files which consist of the OSG scenegraph bits to render the geometry. Occasionally, other geometry files are used and HEV supports any geometry file that OSG can read (e.g. Inventor or OBJ). HEV also supports a custom text-based format, SAVG, that is a simple description language for geometry. The SAVG file format is a historical artifact of the evolution of our system but is still useful as HEV provides a large suite of command-line tools for creating geometry in the SAVG format and modifying SAVG files.

For very common cases, such as volume visualization of spatial volume datasets (e.g. microtomography), there is an IDEA tool that can generate a volume visualization application for the input data. This tool `vol-visBuilder` takes the volume data files and a transfer function description and outputs a set of geometry files, shader sources, textures, and shell scripts which can be immediately viewed. Figure 4 shows an example of the `RgbaLut1D` transfer function description where the volume data set is assumed to have a single scalar value at each voxel. The `VAL_TRANSFORM` keyword specifies a value transform, the `RGBLUT` and `ALPHALUT` tables define the RGB and alpha lookup up tables respectively.

3.2 HEV Tools

After input data file processing, the geometry files can be immediately viewed using `irisfly`. However, a visualization application needs additional functionality to be useful and HEV provides a large set of generic tools for adding this functionality. These tools range from a simple "gnomon" orientation tool, to a "light-editor" that can modify the pre-set lights in the scenegraph, to a geometry clipping control (described above), and a few different animation controllers that have evolved depending on the type of "animation" being used.

These tools all share several common features: 1) they have command-line interfaces, 2) they are small in scope and focus on a single feature, and 3) they communicate by sending their output to the console which is then redirected to the Control Pipe. These three features enable fast iteration in the visualization application development process by providing flexible and composable visualization control, measurement, and analysis tools. The command-line interface enables the tools to be run while the visualization application is under development and also run from shell scripts for additional flexibility. The small scope of each tool ensures that each tool remains simple to implement. Console output allows the tools to interactively modify the in-memory scenegraph through redirection to the Control Pipe.

```
LOAD wirebox wirebox.savg
ADDCHILD wirebox world
EXEC hev-gnomon > $IRIS_CONTROL_FIFO
BACKGROUND 1 1 1
```

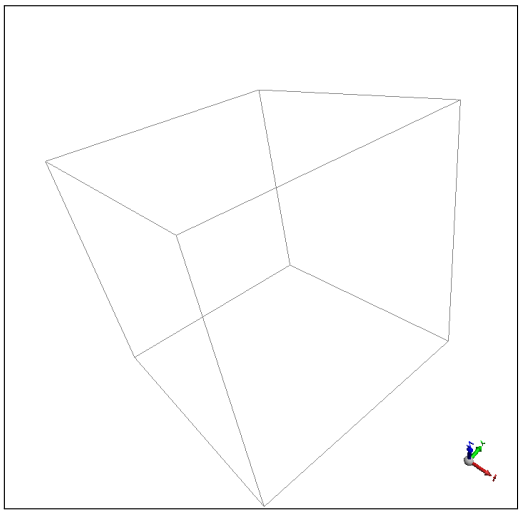


Figure 5: Example of an IRIS file and the associated rendered view.

The development process continues by deciding which of these tools are most useful for the specific visualization application and incorporating them. If a visualization requires new functionality, a new tool can be developed and if it has the above three features, then it will integrate well with the existing HEV tools. As the visualization application is expanded an IRIS file is created that encompasses loading the geometry, running any tools, and adding user interface elements. IRIS files consist of the same text-based control commands for the Control Pipe interpreted by `iris-viewer`. IRIS files are the key component of our application creation framework.

Figure 5 is a basic IRIS file and corresponding rendered view that `LOAD`s a wireframe cube into a node named *wirebox*, adds the node to *world*, `EXEC`s the `hev-gnomon` tool to provide an orientation gnomon in the scene, and sets the `BACKGROUND` of the render window to white. The `$IRIS_CONTROL_FIFO` is a environment variable that specifies the location of the Control Pipe.

3.3 User Interface

As previously mentioned, IDEA uses standard X11 windows for the visualization application GUI. A useful testing utility, `hev-wiggleNav` is a simple GUI tool which causes the virtual environment to rock back and forth, providing enhanced depth cues in desktop windows by means of motion parallax. The tool, when run, displays a window as shown in Figure 7. While running the tool directly is useful for development, a better user experience is provided if the wiggle nav UI can be shown or hidden by the user when necessary. To enable this, the Master Control Panel (MCP) seen in Figure 6 can be extended with additional buttons and menus via MCP files, which are text-based descriptions of UI elements. Figure 8 shows an example MCP file that adds the *wiggle* button to the MCP in Figure 6. When clicked, this button toggles the wiggle nav UI.

There is also a query control command for querying the scenegraph. These query commands allow for more complex tools to interact with IRIS over the Control Pipe. One example that uses the query interface is an interactive "light editor" that enables run-time

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Figure 6: The Master Control Panel (Section 3.3).

Figure 7: The wiggle nav UI (Section 3.3).

```
BUTTON wiggle
  FIRST AFTER REALIZE EXEC hev-wiggleNav \
      -title wiggle > $IRIS_CONTROL_FIFO

  FIRST WAIT irisfly-addAndShowWindow wiggle
  ON WAIT irisfly-addAndShowWindow wiggle
  OFF WAIT irisfly-removeAndHideWindow wiggle
```

Figure 8: The wiggle MCP file (Section 3.3).

editing (adding, removing, orienting, etc.) of the lights in the scene. Figure 9 shows the user interface of this tool.

We have also implemented a server running in Node.js that listens on an HTTP WebSocket and forwards control commands to the control pipe as well as query replies back across the WebSocket. This server has enabled us to develop web-based visualization applications leveraging D3.js running in a web-browser and interacting with the data being rendered in the 3D immersive environment. This can be seen in Figures 14 and 15.

Specifically, in Figure 14, the yellow arrow is the HEV probe tool which updates a shared memory object with the 3D location of the probe. Figure 10 shows the complex MCP file for the probe. In this listing, the `FIRST` commands create the state for the probe when the button is first selected:

1. `probeRun.sh` creates a 2D message box that is updated with the current probe position in 3D space.
2. `irisfly-select` updates a share memory "selector" for other processes to determine when the probe is active.
3. `pointerGlyph.iris` and `plus3d.osg` are the geometries for the probe.
4. the `hev-shmOnOff` commands execute actions for the left and right wand buttons.
5. `irisfly-addAndShowWindow` ensures the probe windows are visible.

The `ON` commands enable all processes on the `probe` selector, enable the geometry node, and show the probe windows. The `OFF` commands reverse the `ON` commands.

This section has only covered a few of the tools available in HEV. As previously mentioned, there is a clipping control tool that supports several different types of clipping geometries and also animation control tools that support a few different types of animations for time-varying datasets. All combined, we have implemented over

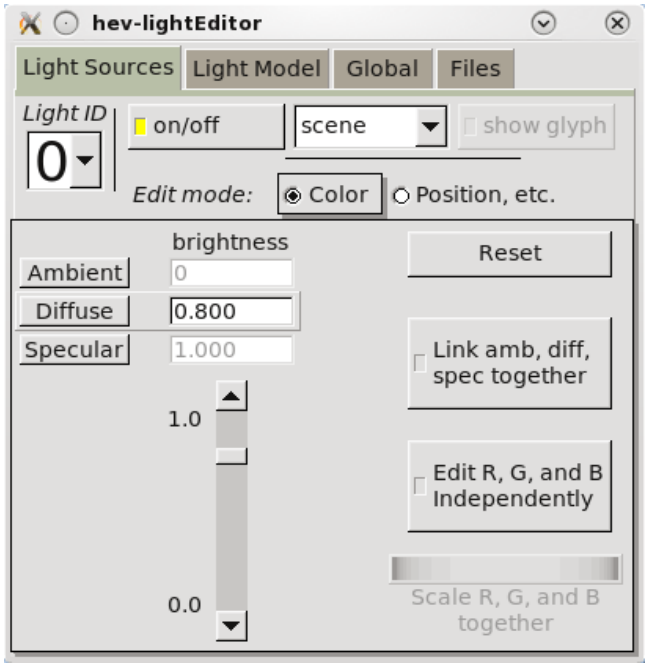


Figure 9: The light editor user interface. This tool enables run-time editing (adding, removing, orienting, etc.) of the lights in the scene.

```
BUTTON probe
  FIRST EXEC $HEV_IDEA_DIR/etc/hev-probe/bin/probeRun.sh
  FIRST WAIT irisfly-select probe

  FIRST LOAD \
    ${HEV_IDEA_DIR}/etc/hev-probe/data/pointerGlyph.iris
  FIRST LOAD irisflyPlus3d plus3d.osg
  FIRST NOCLIP irisflyPlus3d

  # what to do when the wand buttons are pressed
  FIRST EXEC hev-shmOnOff --selector probe \
    idea/buttons/left \
    < $HEV_IDEA_DIR/etc/hev-probe/data/eventLeft.onOff \
    > $IRIS_CONTROL_FIFO
  FIRST EXEC hev-shmOnOff --selector probe \
    idea/buttons/right \
    < $HEV_IDEA_DIR/etc/hev-probe/data/eventRight.onOff \
    > $IRIS_CONTROL_FIFO

  FIRST WAIT irisfly-addAndShowWindow probePosition

  # make the probe visible and turn on button routing,
  # make message windows visible
  ON WAIT irisfly-select probe
  ON NODEMASK irisflyPointerGlyph ON
  ON WAIT irisfly-addAndShowWindow probePosition
  ON WAIT irisfly-addAndShowWindow probe@

  # make the probe and its message windows invisible
  # and turn off button routing
  OFF NODEMASK irisflyPointerGlyph OFF
  OFF WAIT irisfly-removeAndHideWindow probePosition
  OFF WAIT irisfly-removeAndHideWindow probe@
  OFF WAIT irisfly-deselect
```

Figure 10: The probe MCP file discussed in Section 3.3.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

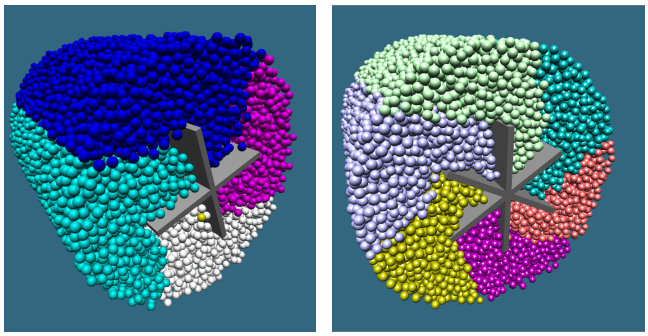


Figure 11: Simulation snapshots of a proposed mortar standard reference material in a 4-blade (left) and a 6-blade (right) rheometer. The spheres are color-coded based on their starting position relative to each vane blade.

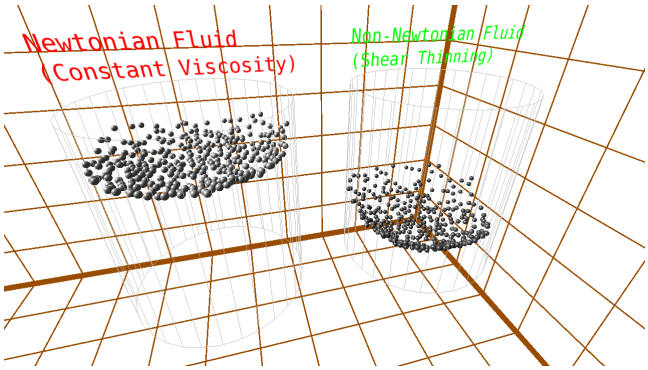


Figure 12: Side-by-side comparison of Newtonian and Non-Newtonian fluids flowing through a pipe. One layer is isolated for visual speed comparison during animation of the simulation data.

250 different tools over the lifetime of HEV. By ensuring each tool is small and focused on a single task and sends its output to the console, these tools are easily composed into larger visualization applications. The next section presents some of these applications that we have developed.

4 Visualization Applications

HEV is currently in active use and visualization applications have been developed over the years. Each application involves at least one domain scientist and the visualization is tailored to the specific scientific data. The applications focus on providing quantitative measurement and analysis of the data.

4.1 Rheology of Dense Suspensions

Understanding the mechanisms of dispersion or agglomeration of particulate matter in complex fluids, such as suspensions, is important in many industries such as pharmaceuticals, coatings, and construction. These fluids are disordered systems consisting of a variety of components with disparate properties that can interact in many different ways. Modeling and predicting the flow of such systems requires large-scale simulations. A large-scale simulation tool [8, 9, 10] is used in the development of standard reference materials (SRM) for the calibration of vane rheometers used to measure the flow properties of fresh concrete. Figure 11 shows snapshots of the rheometer simulations for a proposed mortar SRM.

A second flow of interest to industry is the flow of dense suspensions through a pipe. Flow of suspensions in pipe or channel systems is important for a wide variety of applications including

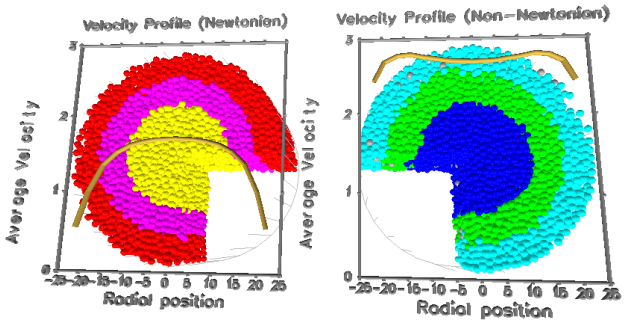


Figure 13: Velocity profile plots show velocity of the solids near the pipe wall is much greater for Non-Newtonian fluids.

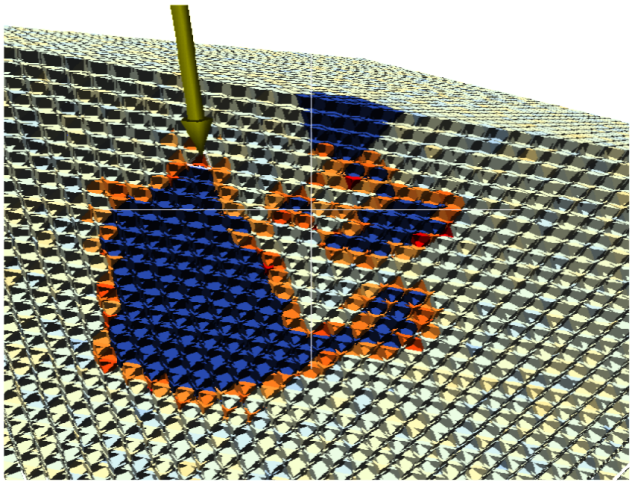


Figure 14: Immersive display of the growth of C3S particles (rendered in blue) hydrating in a calcium hydroxide solution. Regions of orange are where hydration reactions have produced solid calcium hydrate reaction product.

pumping of concrete and slurries, micro-fluidic devices, and biological systems. A visualization application was developed to compare the results from two pipe flow simulations. One data set was for a Newtonian fluid and the other for a Non-Newtonian fluid. Figure 12 shows the side-by-side comparison of the two pipe flow simulations. Controls allow playing the simulation forward and backward. A single layer can be isolated to better compare the velocity of the two data sets. For this visualization application, an analysis tool was developed to show the average velocity of each flow as a line graph. These average velocity graphs are displayed in a heads-up fashion and are dynamically updated corresponding to both the simulation time step and layer position. Figure 13 illustrates this new analysis tool.

4.2 Cement Paste Hydration and Microstructure Development

When cement powder is mixed with water, a hydration process occurs that transforms the cement-water paste from a fluid suspension into a hardened solid. This process involves complex chemical and micro-structural changes. Understanding and predicting the rates of these changes is a longstanding goal. Computational modeling of the hydration of cement is challenging because it involves a large number of coupled non-linear rate equations that must be solved in a highly irregular 3D spatial domain. HydratiCA is a stochas-

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

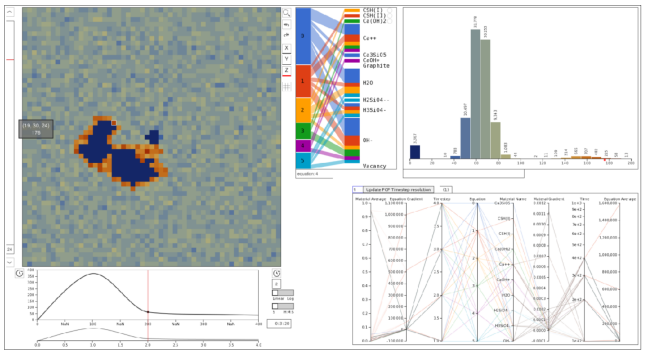


Figure 15: Quantitative graphs of the simulation output corresponding to the 3D visualization in Figure 14. The graphs are dynamically updated as the timestep is changed in the immersive environment.

tic reaction-diffusion cement hydration computational model that addresses these challenges [1, 2, 3, 6].

Figure 14 shows the growth of C3S particles (rendered in blue) hydrating in a calcium hydroxide solution. Regions of orange are where hydration reactions have produced solid calcium hydrate reaction product. This figure is the 3D immersive display of the simulation tool output. For this visualization application, a hybrid tool was developed [5] that provides a 3D immersive view and 2D quantitative view of the same dataset. The 2D quantitative view is implemented in D3 and runs in a web browser. Figure 15 shows the quantitative graphs. Key to this hybrid visualization application is that the immersive 3D display and 2D quantitative display can communicate through the Control Pipe. This enables two-way probing of the data to be dynamically reflected in both displays.

4.3 Body Area Networks

With recent advances in microelectronics, the technology to build very small and extremely low power wearable and implantable devices is clearly within our reach. However, commercial success of this technology depends on the widespread adoption of a global standard for its communication protocol. Knowledge of radio frequency (RF) propagation is a critical step in this process, enabling RF engineers to optimize their physical layer design to achieve better communication performance. Such information is typically gathered by conducting physical experiments and processing the measured data to obtain propagation channel characteristics. Obtaining sufficient data to study various scenarios is difficult for wearable body area network sensors. Moreover, obtaining data through physical experimentation is extremely challenging or impossible for implantable devices. Therefore, a 3D modeling and visualization platform that is capable of emulating physical experiments is required in understanding RF propagation in body area networks [14, 12].

The modeling tool enables researchers to place a custom designed antenna at the desired location of the human body, set the operating frequency of the node, and simulate the RF propagation in and around the human body. The 3D immersive platform, shown in Figure 16 then visualizes the propagation data as a heatmap of signal loss. Several IDEA tools are used to enable probing of the data such as dynamically-updated point values based on wand position or a line-segment tool that uses R to generate summary plots of the data along the line-segment.

5 Conclusion

Content creation is difficult for rendering systems and visualization compounds the problem by needing to visualize widely disparate types of data. In research settings, easy content creation frequently

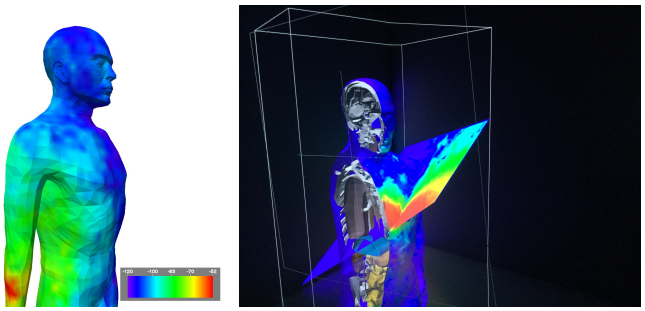


Figure 16: Heatmap of RF propagation signal loss. Propagation on the body (left) and on a plane inside the data volume (right). The right figure also uses clipping planes to show the inside of the body.

drives the choice of rendering and simulation systems. We have developed the High End Visualization (HEV) system that incorporates a visualization application creation framework built on top of a control command interface that can dynamically update an in-memory scenegraph. Small, focused tools that send output to the console are composed using the Control Pipe and IRIS files. This framework has enabled rapid and flexible visualization application development that uses a single rendering executable to run on both a CAVE system and desktop workstation.

We have been migrating our code base to git and will be publishing the software on Github in the near future. We are also evaluating replacing the rendering backend to leverage the new low-overhead graphics APIs such as Vulkan. Before replacing the rendering backend, however, we will implement rendering 'unit tests' to verify that changes made to the rendering system do not have adverse consequences on the rendered output.

Acknowledgements

We would like to thank our scientific collaborators on these projects: Jeffrey W. Bullard, Nicos S. Martys, and Kamran Sayrafian-Pour as well as all of the other NIST staff that have worked with us on these projects. We would also like to thank the reviewers who helped improve this paper with their comments.

DISCLAIMER Certain commercial products are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

References

[1] J. W. Bullard, E. Enjolras, W. L. George, S. G. Satterfield, and J. E. Terrill. A parallel reaction-transport model applied to cement hydration and microstructure development. *Modelling and Simulation in Materials Science and Engineering*, 18(2):025007:1–16, 2010.

[2] J. W. Bullard, E. J. Garboczi, W. L. George, N. S. Martys, S. G. Satterfield, and J. E. Terrill. Advancing the materials science of concrete with supercomputers. *Concrete International*, 33(1):24–29, Jan. 2011.

[3] J. W. Bullard, T. Ley, J. G. Hagedorn, R. Desaymons, W. N. Griffin, J. E. Terrill, Q. Hu, S. G. Satterfield, and P. Gough. Direct comparisons of 3d hydration experiments and simulations. In *6th Advances in Cement-Based Materials Conference*, 2015.

[4] J. E. Devaney, S. G. Satterfield, J. G. Hagedorn, J. T. Kelso, A. P. Peskin, W. L. George, T. J. Griffin, H. K. Hung, and R. Kriz. Science at the speed of thought. In Y. Cai, editor, *Ambient Intelligence for Scientific Discovery*, volume 3345 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2005.

[5] W. N. Griffin, D. Catacora, S. G. Satterfield, J. W. Bullard, and J. E. Terrill. Incorporating d3.js information visualization into immersive

virtual environments. In *Electronic Proceedings of IEEE Virtual Reality (VR) 2015*, 2015.

[6] J. G. Hagedorn, J. W. Bullard, R. Desaymons, , W. N. Griffin, J. E. Terrill, T. Ley, Q. Hu, S. G. Satterfield, and P. Gough. A parallelized numeric model of cement hydration. In *XSEDE 2015*, 2015.

[7] J. G. Hagedorn, J. Dunkers, S. G. Satterfield, A. P. Peskin, J. T. Kelso, and J. E. Terrill. Measurement tools for the immersive visualization environment: Steps toward the virtual laboratory. *Journal of Research of NIST*, 112(5):257–270, 2007.

[8] N. Martys. Study of a dissipative particle dynamics based approach for modeling suspensions. volume 49, 2005.

[9] N. Martys, D. Lootens, W. George, and P. Hebraud. Contact and stress anisotropies in start-up flow of colloidal suspensions. *Physical Review E*, 80, 2009.

[10] N. S. Martys, W. L. George, B.-W. Chun, and D. Lootens. A smoothed particle hydrodynamics based fluid model with a spatially dependent viscosity: Application to flow of a suspension with a non-newtonian fluid matrix. *Rheologica Acta*, 49(10), Oct. 2010.

[11] National Institute of Standards and Technology. SRM Definitions, 2016. http://www.nist.gov/srm/definitions.cfm. Accessed January 8, 2016.

[12] K. Sayrafian-Pour, W. Yang, J. Hagedorn, J. Terrill, K. Yazdandoost, and K. Hamaguchi. Channel models for medical implant communication. *International Journal of Wireless Information Networks*, 17(3–4):105–112, Dec. 2010.

[13] J. E. Terrill, W. L. George, T. J. Griffin, J. G. Hagedorn, J. T. Kelso, M. Olano, A. P. Peskin, S. G. Satterfield, J. S. Sims, J. W. Bullard, J. Dunkers, N. S. Martys, A. O'Gallagher, and G. Haemer. Extending measurement science to interactive visualisation environments. In R. Liere, T. Adriaansen, and E. Zudilova-Seinstra, editors, *Trends in Interactive Visualization*, Advanced Information and Knowledge Processing, pages 287–302. Springer London, 2009.

[14] W. Yang, K. Sayrafian-Pour, J. Hagedorn, J. Terrill, K. Yazdandoost, A. Taparugssanagorn, M. Hamalainen, and J. Iinatti. Impact of an aortic valve implant on body surface uwb propagation: A preliminary study. In *Proceedings of the Fifth International Symposium on Medical Information and Communication Technology (ISMICT)*, Mar. 2011.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Micro-Signatures: The Signatures Hidden in Anomaly Detection Systems

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Abstract

The field of intrusion detection is divided into signature detection and anomaly detection. The former involves identifying patterns associated with known attacks and the latter involves attempting to learn a 'normal' pattern of activity and then producing security alerts when behaviors outside of those norms is detected. The n-grams methodology has arguably been the most successful technique for anomaly detection (including for network packet inspection).

In this work, we identify a new type of intrusion detection that neither uses typical signatures nor is anomaly based (though it is closely related to both). We generate n-grams from both malicious content and Snort signatures and use sets of these 'micro-signatures' to identify attacks. This micro-signature capability arises implicitly when the training sets for n-gram anomaly detection systems are scrubbed of malicious content and thus is not new. It was added explicitly by the seminal Anagram network anomaly approach, but was portrayed as a minor enhancement and its effect was not evaluated. In reproducing the Anagram results we find that for our data, the micro-signatures provide the vast majority of the detection capability. What appears on the surface to be an anomaly detection approach achieves most of its effectiveness from a (sometimes merely implicit) signature subsystem. We furthermore find that these micro-signatures enable highly effective standalone detection systems as well as hybrid micro-signature/anomaly systems that generalize to multiple attack classes.

Our results thus shed new light into the functioning of n-gram anomaly detection systems, reveal the need to evaluate the micro-signature contribution within n-gram anomaly research, and open a new avenue of research into how to best use micro-signatures in future detection systems.

1. Introduction

The field of intrusion detection has been an active area of research since at least the late 1980's [1] [2] [3] and is divided into two areas: signature detection and anomaly detection. Signature based intrusion detection systems (IDSs) identify patterns associated with known attacks. Anomaly based IDSs attempt to learn a 'normal' pattern of activity and then produce security alerts when behaviors outside of those norms is detected.

The n-grams methodology has arguably been the most successful technique for anomaly detection. In the late 1990's, the use of n-grams was discovered to be useful for host based anomaly detection [4]. N-grams are simply a collection of arrays of length *n* obtained by applying a sliding window of length *n* to whatever activity is being monitored (e.g., system calls) [5]. N-grams were first applied to analyze network payloads in the PAYL model [6] in 2004 but were limited to 1-grams, as the number of different n-grams that can be acquired can approach a^n where *a* is the number of characters available (e.g., UTF-8 encoding has 1,114,112 code points [7]). In 2006, the seminal Anagram approach for network packet inspection introduced using an *n* value of greater than 1 by discarding frequency information, accepting a small false positive error, and simply storing the set of acquired n-grams in Bloom filters [8].

In this work, we identify a new type of intrusion detection that uses n-grams but is neither anomaly detection nor does it use typical signatures (although it is closely related to both approaches). We generate n-grams from both malicious content and Snort signatures and use sets of these 'micro-signatures' to identify attacks. The micro-signatures are automatically generated and it is groups of signatures that together detect attacks (as opposed to a single signature mapping to a single attack as in most signature based IDSs). We find that these micro-signatures can be used to create highly effective standalone IDSs or can be coupled with n-gram anomaly detection systems for greater detection scope.

We claim to have 'identified' this approach as opposed to 'invented' because it has always occurred implicitly as a hidden sub-system within n-gram anomaly detection. Whenever one scrubs anomaly detection training data of malicious content, a set of n-grams are removed that then get detected as 'novel' (for those n-grams that don't also occur also in some set of benign training data). The very act of cleaning training data implicitly creates and deploys a set of micro-signatures.

Anagram [8] was the first (and only in our literature search) to include the micro-signatures explicitly as a subcomponent (although the work of [9] includes them in also examining the work of [8]). Doing so enabled them to score the micro-signatures differently than the anomalous n-grams whereas in the implicit approach described above they are scored the same. However, the micro-signature contribution to Anagram was portrayed as a modest enhancement to fine-tune the output of an already high-performing system and its effect was not evaluated.

Here, we reproduce the seminal Anagram results for network anomaly detection (using two Anagram style IDSs and a pure anomaly based IDS) and specifically evaluate the contribution of the micro-signature subcomponent vs. the anomaly detection component. We find that on our data, the micro-signature component performs the vast majority of the detection work with the anomaly component providing a minority input.

The discovery of the effectiveness of the micro-signature component then led us to create standalone micro-signature based IDSs with no anomaly component. To our knowledge, such detection systems have never before been created and tested. We find that this approach has higher overall performance in our experiments compared to the Anagram approaches (albeit by a small margin).

This result does not imply a lack of value to anomaly detection. On the contrary, when the malicious content used to train the micro-signature based component in Anagram did not map well to the set of attacks to be detected in the test set, we see the contribution of the anomaly portion becoming predominant and the micro-signatures playing a supporting role (the converse of what we normally found).

In summary, the primary findings of this paper are the following:

1. N-gram based anomaly detection systems necessarily have two detection components, an anomaly detector and a micro-signature detector.

Harang, Richard; Mell, Peter.
"Micro-Signatures: The Effectiveness of Known Bad N-Grams for Network Anomaly Detection."
Paper presented at 9th International Symposium on Foundations and Practice of Security, Quebec City, Canada. October 24, 2016 - October 25, 2016.

2. The micro-signature detector is a new type of intrusion detection, mixing anomaly and signature based techniques (n-grams, automatically generated signatures, groups of signatures collectively identifying attacks).
3. The micro-signature component performs the vast majority of the detection work in our reproduction of the seminal Anagram experiments.
4. Micro-signature and n-gram anomaly based system effectively co-exist with each component providing majority input in situations where their relevant strengths apply.
5. Micro-signatures can be used to form highly effective standalone IDSs.

These findings impact the area of intrusion detection in the following ways. First, we have increased the understanding of how n-gram anomaly systems works by identifying the two detection components (anomaly and micro-signature). Second, our results indicate that future n-gram research needs to separately calculate the contribution of the anomaly portion vs. the micro-signature portion to provide accurate measurements of performance (e.g., what appears to be an enhancement to anomaly detection in some research may in reality be due simply to the use of a more exhaustive set of micro-signatures). We question how much of published 'anomaly' detection research really is primarily signature based (we truly don't know). Third, we have opened up a new avenue of research (that is neither anomaly detection nor typical signature detection) in how to best optimize and deploy micro-signatures based IDSs.

The rest of this paper is organized as follows. Section 2 discusses our data sets. Section 3 described how we construct the IDSs. Section 4 provides our results and section 5 is a higher level discussion of these results. Section 6 focuses on the impact of our results to the research community while section 7 provides a list of experiments available for future research. Section 8 discusses related work and section 9 concludes.

2. Data Sets

We used three data sources to create training and test sets:

1. From an operational web server, we collected 106,472,207 port 80 requests over 294 hours.
2. From a combination of scanning, vulnerability assessment, fuzzing, and exploit tools that were targeted at a virtual machine running an identical web stack to the operational web server, we collected 393,814 unique port 80 malicious requests.
3. From a recent set of Snort signatures (version 2962 of the community rules) combined with 301 binary malware samples, we collected 24,883,806 bytes.

From these raw data sets, we generate a gold filter, two bad content filters, five normal filters, a web server test set, and a penetration test set (the use of these are explained in subsequent sections). Note that we constructed these filters and test sets following the same process as described in the Anagram experiments that we reproduced [8]. All port 80 data was pre-processed by stripping off IP and TCP headers.

The gold filter is a set of n-grams stored in a Bloom filter that are to represent non-malicious traffic that has been rigorously checked. To create this filter, we used the first 24 hours of the web server traffic after scrubbing it of malicious packets by using automated tools (including both signature-based and anomaly-based tools) as well as human inspection.

The bad content filters are sets of n-grams that represent malicious activity. One bad filter was created from the first 196,907 requests from the exploit tool dataset. The other bad filter was created from the Snort signatures and malware samples. For the Snort signatures, we used all "content" fields at least *n* bytes in length as well as all fixed terms from the "pcre" fields when those terms were at least *n* characters long. Note that generated n-grams were only added to the bad filters if they did not match any n-grams in the gold filter.

The normal filters are sets of n-grams presumed to be non-malicious (to be much larger than the gold filter since the same rigor of checking is not performed). The create these filters, we used the 198 hours of web server requests that followed the 24 hours used for the gold filter (note that 72 hours of this data remain which we use below). To compare training size effectiveness, we created normal filters using 0, 10, 50, 90, 130, 170, and the full 198 hours of this training date. As in [8], we sanitized this training data as follows: we did not insert any n-grams that matched the bad content filter and we did not insert any n-grams from a packet that had 5% or more of its n-grams match the bad content filter.

The Bloom filters used for the above data sets were constructed using a 2^{24} bit index with 3 hash functions per item and using SHA-1 as the hash function, as in [8]. We used an n-gram size of 5 as [8] cited this as being good for 'general' purpose experiments.

The primary test set consisted of 72 hours of unused web server requests. This data was carefully scrutinized using the same method as with the gold filter to label the requests as malicious and non-malicious. 6271 "malicious" packets were found containing a combination of port scans, web server content enumeration scans, SQL injection attacks, and malformed content which appeared to be designed to evoke error messages for service fingerprinting. We refer to this test set as the 'web server' test set.

An additional test set was created from the remaining 196,907 unused port 80 malicious requests taken from the suite of exploit tools. Since this test set consists entirely of malicious requests, it is not used directly. Instead, it is combined with the web server test set to provide what we refer to as the 'augmented' test set.

3. Intrusion Detection System Construction

We used the gold filter, two bad content filters, and seven normal filters along with four different scoring rules to construct a total of 56 different IDSs (1 gold x 2 bad x 7 normal x 4 scoring rules).

To score a particular request, we matched each n-gram against the various filters to produce an ordered tuple (n_1, n_2, n_3) containing a) n_1 as the number of n-grams that matched the normal or gold content filter; b) n_2 as the number of n-grams that matched the bad content filter (referred to as the "micro-signature filter"); and c) n_3 as the number of n-grams that appeared in neither (for the sake of brevity we refer to this last filter as the "novel content filter," however it is never explicitly constructed). It is clear by construction that these three counts are disjoint, their sum is the number of n-grams in the packet. A score for a given tuple is generated by a normalized inner product:

$$S = \frac{\sum_i n_i w_i}{\sum_i n_i}$$

The selection of w corresponds to a particular scoring rule, of which we consider four:

1. The original Anagram scoring rule: $w_1 = 0, w_2 = 5, w_3 = 1$, which we refer to as "Anagram-(0,5,1)"

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2. An unweighted version of the Anagram scoring rule: $w_1 = 0, w_2 = 1, w_3 = 1$, which we refer to as "Anagram-(0,1,1)"

3. A scoring rule which considers only n-grams from the bad content filter: $w_1 = 0, w_2 = 1, w_3 = 0$, which we refer to as "Micro-signature-(0,1,0)"

4. A true anomaly scoring rule which scores on never before seen n-grams: $w_1 = 0, w_2 = 0, w_3 = 1$, which we refer to as "Anomaly-(0,0,1)". Note that to avoid deliberately adding bad n-grams to traffic considered 'normal' under this approach, we used an empty bad content filter. Thus the known malicious traffic was not processed, enabling the related n-grams to be detected as 'novel'. This is a naïve approach as described below and is used to fully explore the set of possible scoring classes.

It is worth noting that – ignoring the magnitude of the weights and restricting them to the same sign – there are 8 possible classes of scoring rules, which can be reduced to 4 by symmetry. For instance, the (1,0,1) rule is simply the complement of the micro-signature scoring rule (0,1,0). The symmetric class pairing are (0,0,0)/(1,1,1), (0,0,1)/(1,1,0), (0,1,0)/(1,0,1), and (1,0,0)/(0,1,1). The class (0,0,0)/(1,1,1) is trivial, always returning a constant value. The class (0,0,1)/(1,1,0) is a true anomaly detector that reveals never before seen n-grams. This is a naïve approach because it ignores the distinction between the gold, normal, and bad content filters and we did not expected it to be useful for intrusion detection (however, see our results). Class (0,1,0)/(1,0,1) represents the micro-signature scoring rule and class (1,0,0)/(0,1,1) represents the Anagram scoring rules. We thus evaluate all available scoring classes.

In our empirical work, each of the 56 IDSs is applied against both the web server test set and the augmented web server test set for a total of 112 experiments.

4. Results

We compare our IDSs using the usual Receiver Operating Curves. In particular, we focus on the area under the curve (AUC) to compare true positive performance across a wide range of false positive rates.

Table 1 provides a high level comparison, showing the mean AUC for each of the four scoring rules with 90 hours of training data for the normal filter using both the web server test set and the augmented test set along with both bad content data sets. A value of 1.0 indicates perfect classification (a true positive rate of 1.0 may be obtained with a false positive rate of 0) while a value of 0.5 is the value that can be obtained by random guessing.

Table 1. Mean AUC Across Both Test Sets

Intrusion Detection System	Mean AUC
Anagram-(0,1,1)	0.93
Microsig-(0,1,0)	0.94
Anagram-(0,5,1)	0.91
Anomaly-(0,0,1)	0.74

Note how the Micro-signature approach performed equivalently to both Anagram approaches. This means that using the n-gram signatures alone produces comparable overall performance to using the n-gram signatures in conjunction with anomaly detection. Note that while the performance of the Micro-signature approach slightly exceeds the Anagram approaches here, the point of this research is not to identify a better IDS, but to show three things: 1) how micro-signatures have been providing the majority portion of the performance of Anagram based IDSs relative to our datasets, 2) the process of filtering out bad content from training traffic implicitly creates these micro-signatures and thus their use is almost unavoidable, 3) micro-signatures can be used to create effective standalone IDSs.

The pure anomaly detection approach performed much worse. However, this was not surprising as it was truly just detecting on never before seen n-grams. We did not filter out bad n-grams embedded in the training set as doing so would have implicitly moved such bad n-grams into the novel content filter (filter n_3 in section 3), thereby converted the pure anomaly detector into having similar results as Anagram-(0,1,1), which we already evaluate. Notice how the very act of trying to filter out bad content from the anomaly detection system's training set implicitly creates a micro-signature detection capability (which we find in this research to be extremely powerful and never before analyzed in the literature).

One might be tempted from these results to discount anomaly detection altogether and simply rely on micro-signatures. However, we find that in circumstances where the micro-signatures do not correlate well to the attacks in the test set, that the anomaly portion of Anagram automatically jumps in an plays a majority role in detection. Overall though, our data indicates that anomaly detection provides a supporting role to micro-signature detection, which does the vast majority of the detection work. The existing literature (see [8]) asserts the opposite without ever explicitly analyzing the contribution of the micro-signatures. Note that we aren't claiming that our results generalize here to other data sets, but our counter examples demonstrate the need for research efforts to document the contribution of both subsystems.

In the next two sub-sections, we evaluate the AUC results for the four scoring methods using differing combinations of test and training sets and differing amounts of training data. After about 90 hours of training data, the AUC values remain high and relatively stable with respect to the amount of training data (broadly consistent with the findings of [8]). However at lower amounts of training data the behavior becomes slightly less consistent. The behavior of the models with no training data whatsoever is surprising but illuminating, as we discuss in more detail below.

4.1 Results for the Web Server Test Set

We now look in more detail at the results for the web server test set. In Figure 1, we evaluate IDS performance with varying sized training sets for the normal content filter and use the web server exploit tool training content for the bad content filter.

The Micro-signature approach provides slightly better performance (AUC of .987) than the Anagram combined micro-signature/anomaly approaches. Note that the micro-signature set was trained against a set of web exploits and so the trained signature set is appropriate for the target set of attacks to be detected (web server attacks). This likely explains the high performance of the methods using micro-signatures (the Micro-signature and Anagram approaches).

The anomaly approach did not do any better than random except with 0 training data for the normal content filter. This data outlier reflects an anomaly system where only the gold filter was used as normal and all else was flagged as novel. The gold filter was carefully created by scrubbing it of all malicious n-grams thereby implicitly adding the malicious n-grams to the novel content filter. In this special case, the Anomaly-(0,0,1) then acts as a micro-

signature detector and gains an enormous performance enhancement. The Anagram-(0,1,1) acts identically here (with an AUC of .963) but divides the micro-signatures and novel n-grams between two equally weighted sets. The Anagram-(0,5,1) rule has slightly worse performance (AUC of .958) because of the unequal 5 to 1 weighting of micro-signatures and novel n-grams.

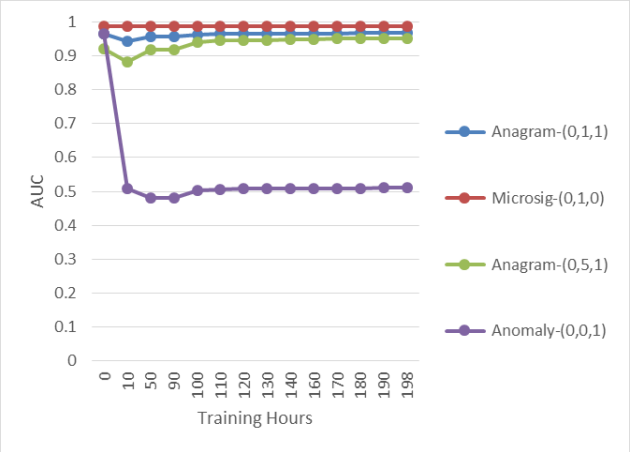


Figure 1. Web Server Test Set with Web Server Exploit Tool Training Content

To look deeper into how micro-signatures contribute to anomaly detection systems in this scenario, we evaluate the relative contribution of novel n-grams vs. micro-signatures within the Anagram-(0,1,1) approach. To do this, we plot in Figure 2 each point correctly classified as 'malicious' by Anagram-(0,1,1) on the x-y plane with the x coordinate being the portion of the score attributable to the micro-signatures and the y coordinate being the portion of the score attributable to the novel content filter. Note that because these two sets are mutually exclusive, all points will lie in the region $\{x > 0; y > 0; x + y < 1\}$. A kernel density estimate to help visualize the distribution of the points is overlaid. The points themselves are plotted with an alpha of 0.05 over the graph; the dashed line indicates equality. In this case (and for all future such plots) we used 130 hours of training data for the normal filter.

A significant number of points in both plots lie along the y=0 line, indicating that none of the n-grams leading to the malicious classification were found in the novel content filter. By contrast, virtually no points lie along the x=0 line. Additionally, the bulk of the density of the distribution lies firmly below the diagonal line, indicating that the majority of the score for the most of the packets was derived from the micro-signature filter, which we thus conclude is doing the 'heavy lifting'.

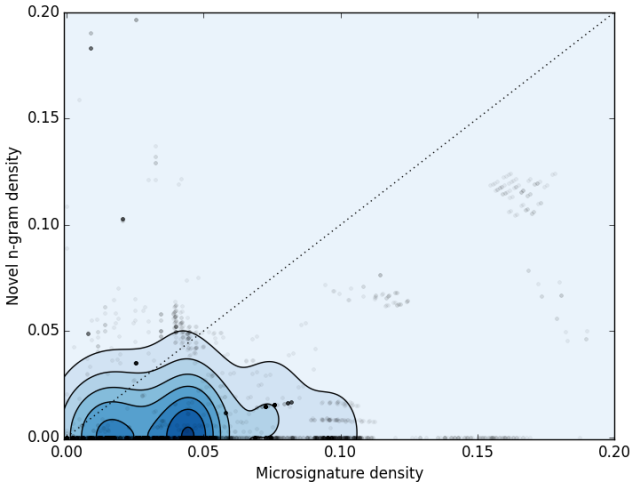


Figure 2. Anagram-(0,1,1) Relative Contribution of Anomaly Detection vs. Micro-Signature Detection for the Web Server Test Set with Web Server Exploit Tool Training Content

We now re-run the same experiment except this time we use the Snort/malware training set (which is not focused on the web traffic being tested) for the micro-signature filter. Figure 3 shows the results. Note the degraded performance of the Micro-signature approach, apparently due to the signature set not aligning as well with the set of attacks to be detected.

However, the Anagram approaches also suffer degraded performance. In part, (shown below) this is because they also rely heavily on the micro-signature filter. But also, consider the consequence of having an ineffective micro-signature filter during training of the anomaly detection capability. If our micro-signatures alone cannot already detect attacks in the training set, then the normal content model that will be constructed will inevitably contain some traffic from malicious packets. This in turn will lead to similar malicious packets being judged to be "more normal" which in turn will lead to a higher rate of false negatives.

Again, the outlier point with 0 training data for the Anagram and Anomaly approaches demonstrates the strength of the micro-signatures. During training, the gold filter was scrubbed of malicious web server traffic n-grams and these micro-signatures were implicitly added to the novel content filter enabling the Anagram and Anomaly approaches to act to a large degree as signature systems.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

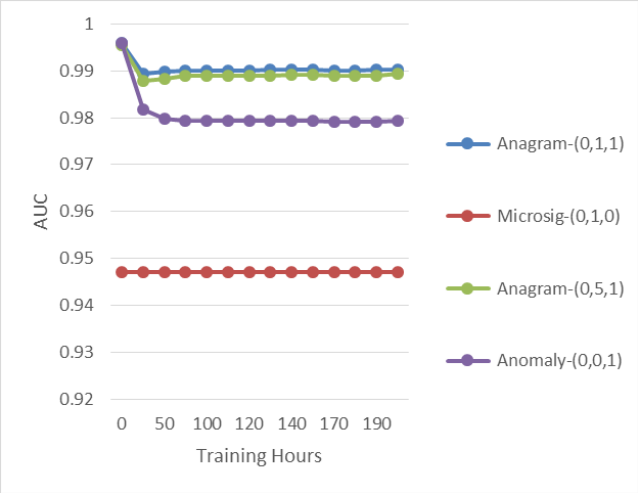


Figure 3. Web Server Test Set with Snort/Malware Training Content

Figure 4 shows the relative contribution of novel n-grams vs. micro-signatures within the Anagram-(0,1,1) approach for this experiment. While the micro-signatures are still doing the majority of the detection work and there are many points on the x-axis (denoting no contribution by the novel component), note that the novel n-grams contribute more than in Figure 2. We attribute this to the anomaly system helping out more when the generated micro-signatures are less appropriate for the attack domain to be detected.

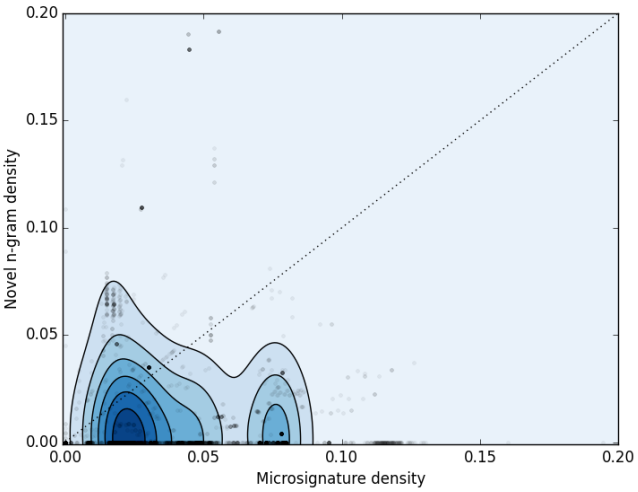


Figure 4. Anagram-(0,1,1) Relative Contribution of Anomaly Detection vs. Micro-Signature Detection for the Web Server Test Set with Snort/Malware Training Content

4.2 Results for the Augmented Test Set

We now look in detail at the results for the augmented test set. Recall that this test set is the web server test set augmented with an additional 196907 malicious requests generated from exploit tools. Given that only 6271 malicious requests were in the web server test set, the overwhelming majority (97 %) of the malicious requests in this augmented test set came from the exploit tools.

In Figure 5 we see that the performance of the Micro-signature and Anagram approaches are similar to that with the web server test set. However, the anomaly approach has improved from performing randomly to obtaining an AUC of almost .98. This increase is explained by noting that the Anomaly approach was only exposed to 6271 malicious requests during training which was not sufficient to prevent it from detecting the 196907 exploit requests as novel (i.e., the micro-signature sets were sufficiently distinct). We posit that had the n-grams generated from the malicious requests in the training data provided more coverage of the malicious requests in the test set, the performance of the anomaly system would have been much worse.

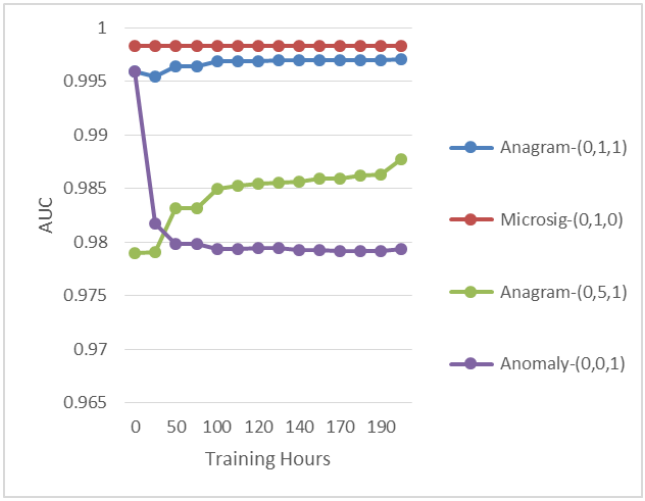


Figure 5. Augmented Test Set with Web Server Exploit Tool Training Content

Figure 6 shows the relative contribution of novel n-grams vs. micro-signatures within the Anagram-(0,1,1) approach for this experiment. Note that to permit a clear visualization of the contours we have omitted the individual points. Here the training set for the micro-signature filter most closely matched the malicious requests in the test set (recall that the web server exploit tool malicious requests were divided equally into a set used for training and a set used for testing).

Note that the novel n-gram density is extremely low compared to the other plots and the micro-signature density is so high that we had to change the x-axis scaling compared to the other graphs just to make the data visible. This can be explained by noting that the micro-signature filter so closely covered the malicious requests in the test set that there were few unmatched malicious n-grams left to label anomalous. We will see the reverse phenomenon happen in the next pair of figures where the micro-signatures do not correspond well to the malicious requests in the test set.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

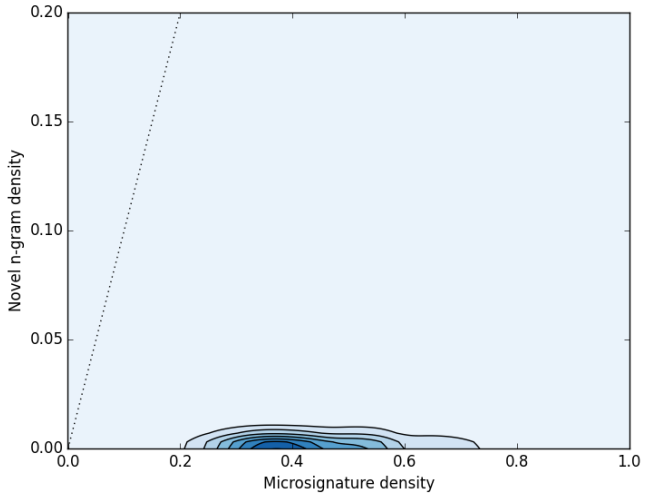


Figure 6. Anagram-(0,1,1) Relative Contribution of Anomaly Detection vs. Micro-Signature Detection for the Augmented Test Set with Web Server Exploit Tool Training Content

We now re-run the same experiment except this time using the Snort/malware training set for the micro-signature filters. Figure 7 shows how for the first time in our experiments, the Micro-signature approach performs worse than the other approaches, albeit by a small margin (note the scaling). The AUC difference between the top performing Anagram-(0,1,1) and the Micro-signatures at 90 hours of training is just .043 and the Micro-signature approach still achieves an AUC of .95. The drop in performance is analogous to the Micro-signature performance drop from Figure 1 to Figure 3. As in this former case, our analysis indicates that the reason is that the Snort/malware training set is less suitable for generating signatures for web server attacks than the web server exploit tool training set.

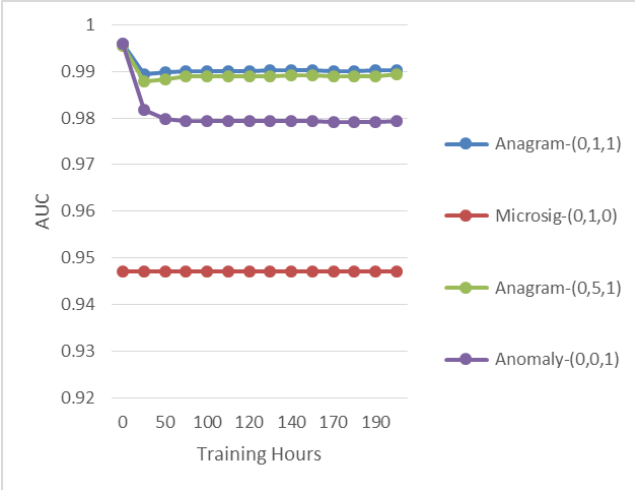


Figure 7. Augmented Test Set with Snort/Malware Training Content

As done previously, we now plot in Figure 8 the relative performance of the novel n-grams vs. the micro-signatures for the Anagram-(0,1,1) approach. We see for the first time the novel n-grams playing a larger role than the micro-signatures. This effect (the converse of that shown in Figure 6) was expected as the micro-signatures generated from the Snort/malware training set poorly matched the test set of malicious web server requests. As a result, the detection burden automatically shifted to the novel n-grams which demonstrates the flexibility of the hybrid micro-signature/anomaly capability within the Anagram approaches. Note, however, that even here the micro-signatures do play a significant role whereas in the converse case of Figure 6, the novel n-grams played almost no role (note the difference in the y-axis scaling of both plots).

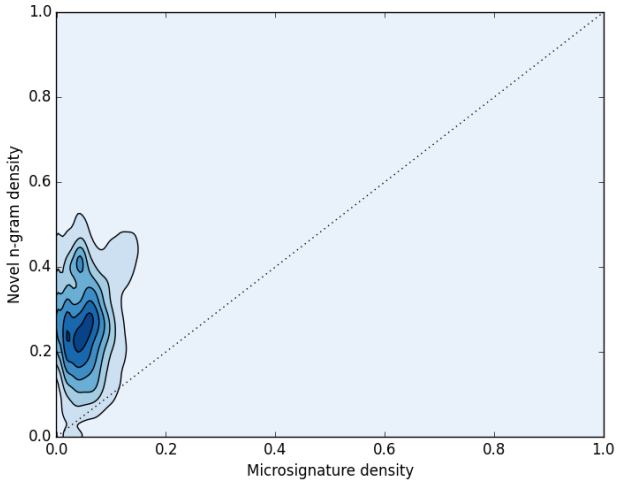


Figure 8. Anagram-(0,1,1) Relative Contribution of Anomaly Detection vs. Micro-Signature Detection for the Augmented Test Set with Snort/Malware Training Content

5. Discussion

Micro-signatures are not a new discovery (having been included within Anagram in 2006), but they were seen as a minor contributor and were not separately evaluated.

Our work reproduces and expands upon the seminal Anagram experiments in [8]. We show that the Anagram approach is clearly effective for HTTP requests. More significantly, we provide the first study that analyzes the contribution to detection made by the subcomponents of Anagram (separating out the anomaly portion from the micro-signature portion). Quite surprisingly, we find that for our data the micro-signatures portion contributed much more to the detection capability than the anomaly portion. This means that, relative to our datasets, the seminal Anagram anomaly detection system that proved the usefulness of n-grams for network packet inspection achieves the majority of its effectiveness from a subsystem that is effectively signature based.

However, this signature based subsystem is very different from typical signature based systems. The signatures are automatically generated from known malicious packets and are very small in size (sets of 5 characters in our experiments). It is the presence of groups of signatures that are indicative of an attack, not just single signatures as is the case with standard signature based IDSs. Because each signature is less focused on a single attack, the signatures appear in our data to generalize to new attacks within the same attack class.

Interestingly, the use of micro-signatures is almost unavoidable for n-gram based anomaly detection systems. Any time training data is filtered of malicious content, the filtered n-grams (unless they also appear elsewhere within unfiltered non-malicious training content) are implicitly forced into the novel content filter. Virtually all n-gram based anomaly systems then benefit from micro-signatures without ever explicitly using them. More thorough and accurate scrubbing of the training data will produce more thorough and accurate micro-signatures. Many papers that have focused on

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

"anomaly detection" using training data that has been scrubbed of malicious content (virtually all of them) have – in effect – been relying heavily on signature-based methods despite being termed "anomaly detection". We make no claim that for other n-gram anomaly systems and datasets we will see the same relative contribution of the components (although we suspect this to roughly true), but one major point of our research is that the relative contribution is important and should be measured.

Another new discovery of this work is that the micro-signatures can be effective on their own, apart from being coupled with an anomaly detection system. Surprisingly, they can function better than the Anagram hybrid micro-signature/anomaly method. That said, we showed how in cases where the attacks in the training set do not correspond well to the attacks in the test set, the anomaly portion of the hybrid approaches kicks in to boost the performance above that of the micro-signatures alone. This leads to the observation that hybrid systems using both micro-signatures and anomaly approaches provide a broader scope in detecting varying classes of attacks.

Another unexpected result was that at 0 hours of training data for the normal filter the anomaly systems performed extremely well. In fact, they often performed the best with 0 hours of training data for the normal filter. We explain this by noting that with 0 hours of training data for the normal filter, the anomaly algorithms are only using the n-grams from the gold filter that were taken from 24 hours of highly scrubbed web server requests. This scrubbing implicitly created micro-signatures that enabled the high detection rate. Our conclusion here is that a smaller amount of carefully scrubbed training data can create a more effective hybrid micro-signature/anomaly detection system than one with a larger amount of less carefully scrubbed data.

As a final issue, we consider the resilience of micro-signatures to evasion attacks. In particular, the normalization to packet length in our Micro-signature approach could lead to an evasion attack where a malicious packet is stuffed with a lot of normal data; this "content mimicry" attack is considered within the original Anagram paper, where it is addressed via subsampling of the packet payload [8]. While the mimicry resistant approach suggested in the original Anagram paper will likely not be as effective for micro-signatures, another potential avenue for handling content mimicry might be through not normalizing the micro-signature counts to packet length. Not shown in this paper are results which find that this idea is effective, but has worse performance than normalized micro-signatures.

6. Impact of Results

How do our results impact the field of intrusion detection? This is an especially valid question since micro-signatures are already being used, albeit unknowingly in virtually all n-gram based anomaly detection. In addition, they were even used explicitly in Anagram, although not evaluated for effectiveness and assumed to be a minor contributor.

One answer is that our results provide us a new understanding of how n-gram anomaly detection functions. We now understand that n-gram anomaly detection systems almost unavoidably contain a signature component (whether realized implicitly or explicitly). When cleaning training traffic of malicious content, if the related n-grams are stored in a bad content filter then they can be used explicitly for micro-signature detection. If they are not stored then the related micro-signatures become implicitly added to the novel content filter. Note that this novel content filter is not usually explicitly created but is the set of n-grams not present in the 'normal' traffic filters. If an n-gram anomaly system attempts to avoid using micro-signatures by not cleaning the training traffic, the performance declines drastically as was seen for our Anomaly-(0,0,1) approach. One could argue that instead of scrubbing malicious data from the training set, that a natively clean training set could be provided to avoid creating micro-signatures. A problem with that approach is that such natively clean data sets are usually created in a laboratory setting and often don't represent the variety seen 'in the wild' (thus much real normal traffic will be considered novel). Finally, we see no compelling reason for n-gram anomaly detection systems to attempt to avoid micro-signature use given there benefit. In our data they provided the majority of the detection capability to the hybrid micro-signature/anomaly detection approaches.

The realization that n-gram anomaly detection invariably contains two components impacts how future n-gram approaches should be measured; the contribution of the micro-signature and anomaly components should be explicitly measured. By doing this, researchers will be able to discover whether or not their new technique is an advance in anomaly detection or that it simply uses a better or more focused set of micro-signatures. Measuring the relative contribution of the two sets is not hard, but it does require the researchers to keep track of the filtered n-grams during the process of cleaning the training sets of malicious data (only including those not found normal in other benign training data). Another reason to measure this in future research is to determine the overall contribution of the two components over more varied sets of data. Our experiments showed the micro-signature contribution dominating for our web traffic dataset. It is not yet known whether or not this result generalized to other data sets. Our current hypothesis based on the results of this paper, completely counter to that of the current understanding, is that n-gram anomaly detection is primarily a signature based approach that is only augmented by anomaly detection. We could be completely wrong, however, finding the answer is important regardless of the discovered result. Only a collection of future studies in a variety of domains will determine this overall trend.

Lastly, our results impact the field of intrusion detection by opening a new avenue of research into a new type of intrusion detection (albeit one closely related to both anomaly and traditional signature detection). This includes exploring using micro-signature IDSs as standalone systems as well as in hybrid systems that combine micro-signature and anomaly detection. While micro-signatures have been implicitly used since n-gram anomaly detection was developed, they have never been carefully studied. By deliberately focusing on their development, we will see how far the approach can be optimized and hope that it will lead to deployable systems. N-gram anomaly detection, despite its success in research circles, has not (to our knowledge) been widely deployed commercially due to unacceptable false positive rates. Perhaps the micro-signature system, being closer to traditional signature based approaches but using n-grams like anomaly systems, will have greater operational applicability and be a stepping stone towards enabling the enterprise deployment of anomaly based approaches.

7. Future Work

This section contains a variety of ideas for additional research in this area. The authors do not have the resources to explore all of these and encourage the community to help fully develop this new research area.

Future work should explore how to most effectively use micro-signatures and how to obtain the best accuracy. The various

parameters that can be set for the micro-signatures, including the length of the n-gram used, the parameterization of the Bloom filter (or other data structure), and methods for selecting the threshold parameter in the absence of extensive validation data, all require further study. Future work should explore how well micro-signatures generalize to different types of attacks and never before seen attacks (both within a specific attack class and between different attack classes). The possibility of combining micro-signatures covering different protocols within a single, larger Bloom filter should be explored. Another possible experiment is to create micro-signatures from standard IDS signatures and compare their performance (we expect the micro-signature variant to generalize while the standard signatures will not). A further study can evaluate the extent to which a group of micro-signatures can hinder an attacker from creating variations of attacks that evade current signature sets.

Future work should also be conducted in how to best leverage micro-signatures within n-gram anomaly detection systems. Previously, the micro-signatures were present and used in such systems. Now that we know of their presence, we can research how to best optimize their use in conjunction with the anomaly detection component. One area is to determine the optimal scoring weights for micro-signature and anomaly n-grams for different data sets, or examine alternate ways of deriving more expressive features from them. We saw in our work that the Anagram weighting of 5 for micro-signatures and 1 for novel n-grams performed worse than an equal weighting but no further work was done in this area. Another important step in researching hybrid micro-signature/anomaly systems is in confirming or refuting our conjecture that n-gram anomaly detection in general is primarily a signature based approach. This will likely need to be done by many researchers in different areas testing their unique datasets. While we would have liked to do that within this research, such a breadth of test data is not available to us or (to our knowledge) any single group of researchers. The work of [9] has examined several features of the distribution of n-grams for "normal" packet content that give an indication of how effective n-gram methods are likely to be on such content; it seems worthwhile to explore methods to adapt their measures to malicious content.

Lastly, network forensic related studies should be conducted on how to link micro-signature detected attacks with the relevant source material. This would include identifying the relevant portions of the flagged packets and/or a set of simple signatures (on which the matched micro-signatures were based).

8. Related Work

Given that our work is the first study on micro-signatures, for this related work section we focus on references to n-gram anomaly detection and more general challenges to anomaly detection in the field of machine learning.

The difficulty of applying machine learning in general to intrusion detection is discussed by Sommer and Paxson [12]. They point out several features of intrusion detection problems that make it difficult to successfully apply machine learning. In particular, the rareness of attacks, the high cost of diagnosing detected attacks (particularly when there is a mismatch between the information that a machine learning system provides the analyst and the way in which an analyst diagnoses an event), and the complexity of the input data all mean that machine learning IDS solutions must achieve extremely low error rates on extremely complex problems to be operationally effective. A more probabilistic argument is made in [10] in terms of the base rate fallacy. Nevertheless, multiple examples of anomaly-based and unsupervised network intrusion detection methods can be found in the literature.

One of the earliest n-gram approaches is that of the PAY-L system [6], which clusters network traffic based on the distribution of 1-grams. The Anagram system [8], which forms the basis of our analysis, extends the length of the n-grams to between 5 and 9, while also addressing the issue of "content mimicry". In perhaps the most general case, the issue of anomaly detection via n-grams in non-textual, binary protocols is considered by Hadžiosmanović et al. [9], building on the work of [6] and [8]. This work examines classifiers that make no use of any protocol-specific domain knowledge, and concludes that n-gram based methods generally perform poorly for binary protocols, with an unavoidable tradeoff between high detection rate and low false positive rate. This poor performance relates directly to the 'variability' of the normal traffic (more precisely, the degree to which the n-grams appear to be sampled approximately uniformly from the space of all possible n-grams in normal traffic). While they do not specifically address compressed or encrypted protocols, it seems clear that these will have similar issues. More recently, the work of [9] explores various statistical measures relating to the distribution of n-grams, and relates these measures to the performance of n-gram based supervised and unsupervised classifiers; their work emphasizes machine learning aspects more heavily than our basic analysis, and uses richer feature vectors and more sophisticated classifiers. Similarly to the clustering described in [6], the work of [13] examines the use of a self-organizing map for on-line clustering of packets.

Domain-specific knowledge, in the form of partial parses of protocols, can be used to extract more specific sets of features that help in the identification of anomalous content. In Robertson et al. [14], for instance, web requests are processed by specializing to particular web components, and then extracting key-value pairs from the URIs specific to those components. They learn specialized models (such as simple regular expressions, often simply representing the allowed characters) conditional on each field and component – in effect learning a mixture of site-specific 'sub-protocols' within HTTP. Guangmin [15] performs similar tokenization for use in an artificial immune system model. Ingham et al. [16] attempts to learn deterministic finite automata (DFAs) for normal HTTP traffic while detecting, parsing, and transforming known features (such as email addresses) in order to control complexity. The high degree of structure in the underlying grammar (HTTP) combined with the generally limited character set all contribute to the ability of such systems to be effective. However, these systems are also highly specialized to their particular domain of application and so cannot extend to more general intrusion detection scenarios.

Finally, as machine learning techniques have developed, anomaly-based IDS work has kept pace. More advanced approaches to the problem include that of Gornitz et al. [17]. Here, active learning is used to request that specific packets be labeled by an outside mechanism (e.g. a human analyst) thus maximizing the discriminative power of the learning algorithm within a limited budget of time and effort. While such systems do require more resources to train initially, they typically result in significantly improved performance over purely unsupervised systems. The use of the bad content filter in the Anagram system [8] may be viewed as a non-active, simplified version of this semi-supervised approach.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

9. Conclusion

In reproducing the seminal Anagram research for network anomaly detection, we have identified the important role of micro-signature based intrusion detection. We explored how micro-signature detectors are a new type of intrusion detection, mixing anomaly and signature based techniques (n-grams, automatically generated signatures, and groups of signatures collectively identifying attacks). We furthermore discovered that n-gram based anomaly detection systems necessarily have two detection components, an anomaly detector and a micro-signature detector. We found that for our data the micro-signature component performs the vast majority of the detection work but that the anomaly detection component is still important and a significant contributor. On that point, we find that micro-signature and n-gram anomaly based systems effectively co-exist with each component providing majority input in situations where their relevant strengths apply. Finally, we discover that micro-signatures can be used independently to form highly effective standalone IDSs.

Our discoveries are important in several areas. From a foundational point of view, they provides us a new understanding of how n-gram anomaly detection functions. From an anomaly detection research point of view, they leads us to recommend that all future n-gram anomaly detection research calculate the relative contribution of novel n-grams vs. micro-signatures in order to accurately measure the effectiveness of the anomaly detection. From an operational point of view, they lead us to investigate how to best deploy micro-signatures to augment existing intrusion detection systems. Overall, our results provide the initial discovery of a new area of intrusion detection that is neither standard signature detection nor anomaly detection, opening up a new avenue for IDS research.

10. Works Cited

[1] S. E. Smaha, "Haystack: An intrusion detection system," in *Aerospace Computer Security Applications Conference*, 1988.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. 2, pp. 222-232, 1987.

[3] H. S. Vaccaro and G. E. Liepins, "Detection of anomalous computer session activity," in *IEEE Symposium on Security and Privacy*, 1989.

[4] S. Forrest, S. Hofmeyr and A. Somayaji, "Computer immunology," *Communications of the ACM*, vol. 40, no. 10, pp. 88-96, 1997.

[5] D. Damashek, "Gauging similarity with n-grams: language independent categorization of text," *Science*, vol. 267, no. 5199, pp. 843-848, 1995.

[6] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, Heidelberg, 2004.

[7] "The Unicode Standard Version 6.0- Core Specification," February 2011. [Online]. Available: http://www.unicode.org/versions/Unicode6.0.0/ch01.pdf.

[8] K. Wang, J. J. Parekh and S. J. Stolfo, "Anagram: A content anomaly detector resistant to mimicry attack," in *Recent Advances in Intrusion Detection*, Heidelberg, 2006.

[9] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon and S. Etalle., "N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols," in *Research in Attacks, Intrusions, and Defenses*, 2012.

[10] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Transactions on Information and System Security* , vol. 3, no. 3, pp. 186-205, 2000.

[11] R. Chang, R. E. Harang and G. S. Payer, "Extremely Lightweight Intrusion Detection (ELIDe)," Army Research Laboratory, 2013.

[12] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Security and Privacy* , 2010.

[13] D. Bolzoni, E. Zambon, S. Etalle and P. Hartel, "Poseidon: A 2-tier anomaly-based intrusion detection system," arXiv preprint cs/0511043 , 2005.

[14] W. Robertson, G. Vigna, C. Kruegel and R. A. Kemmerer, "Using generalization and characterization techniques in the anomaly-based detection of web attacks," in *NDSS*, 2006.

[15] L. Guangmin, "Modeling Unknown Web Attacks in Network Anomaly Detection," in *Third International Conference on Convergence and Hybrid Information Technology*, 2008.

[16] K. L. Ingham, A. Somayaji, J. Burge and S. Forrest, "Learning DFA representations of HTTP for protecting web applications," *Computer Networks* , vol. 51, no. 5, pp. 1239-1255, 2007.

[17] N. Görnitz, M. Kloft, K. Rieck and U. Brefeld, "Active learning for network intrusion detection," in *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*, 2009.

[18] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," 2000.

[19] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks* , pp. 2435-2463, 1999.

[20] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks," in *LISA*, 1999.

[21] K. Rieck and P. Laskov, "Detecting unknown network attacks using language models," in *Detection of Intrusions and Malware & Vulnerability Assessment*, 2006.

[22] K. Rieck, P. Laskov and K.-R. Müller, "Efficient algorithms for similarity measures over sequential data: A look beyond kernels," *Pattern Recognition*, pp. 374-383, 2006.

[23] C.-C. G. F., A. Stavrou, M. E. Locasto and S. J. Stolfo, "Adaptive anomaly detection via self-calibration and dynamic updating," *Recent Advances in Intrusion Detection*, pp. 41-60, 2009.

[24] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto and W. Lee, "McPAD: A multiple classifier system for accurate payload-based anomaly detection," *Computer Networks* , vol. 53, no. 6, pp. 864-881, 2009.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Methodology for Increasing the Measurement Accuracy of Image Features

Michael Majurski, Joe Chalfoun, Steven P. Lund, Peter Bajcsy, and Mary Brady
National Institute of Standards & Technology
100 Bureau Drive, Gaithersburg, MD 20899
{michael.majurski, joe.chalfoun, steven.lund, peter.bajcsy, mary.brady}@nist.gov

Abstract

We present an optimization methodology for improving the measurement accuracy of image features for low signal to noise ratio (SNR) images. By superimposing known background noise with high quality images in various proportions, we produce a degraded image set spanning a range of SNRs with reference feature values established from the unmodified high quality images. We then experiment with a variety of image processing spatial filters applied to the degraded images and identify which filter produces an image whose feature values most closely correspond to the reference values. When using the best combination of three filters and six kernel sizes for each feature, the average correlation of feature values between the degraded and high quality images increased from 0.6 *(without filtering) to* 0.92 *(with feature-specific filters), a* 53% *improvement. Selecting a single filter is more practical than having a separate filter per feature. However, this results in a* 1.95% *reduction in correlation and a* 10% *increase in feature residual root mean square error compared to selecting the optimal filter and kernel size per feature. We quantified the tradeoff between a practical solution for all features and feature-specific solution to support decision making.*

1. Introduction

Image features are computed in cell biology to extract quantitative information regarding cell state, differentiation, biological activity, and cell dynamics. The motivation for our work is the improvement of measurement accuracy for image features extracted from time-lapse fluorescent images of stem cell colonies. Due to cell sensitivity to light, only brief low intensity light could be used to excite fluorophores, producing images with low signal to noise ratios (SNRs) and hence resulting in questionable accuracy of image features.

Our objective is to mitigate the effects of image noise on extracted features via image de-noising (filtering) with respect to quantitative metrics. Quantitative imaging can play an important role in scientific experiments as a means to monitor and communicate behavior of complex systems (e.g. cell colonies) by recording features extracted from objects of interest. An image feature is a function whose input is an image. Ideally, the image itself is representative of the current state of the biological system being imaged such that changes in the images are representative of changes in the system. In such cases, image features can provide useful summaries to help monitor and communicate the systems behavior. However, when images have a low SNR, poor focus, or other distortions, the information extracted via feature evaluation may not reflect the behavior of the underlying system. That is the ability to extract meaningful image feature values is linked to the quality of the acquired images. Unfortunately, due to experimental constraints ideal high quality images can be time consuming to acquire, impractical, expensive, or damaging to the specimen. This forces the acquisition of lower quality images. Image processing algorithms can help mitigate measurement inaccuracies caused by low quality images. The difficulty lies in selecting which image processing algorithms to apply for a given feature measurement. We are interested in determining the ordered set of image processing operations that result in images whose feature values convey similar meaning to those of the same image if it was of higher quality, in other words, having clear signal and negligible noise.

In this paper, we focus on the effects of image noise, as opposed to other factors that may degrade image quality. Image features measured from images with low SNRs can be very poorly correlated with ground truth feature values, defined as those extracted from the same signal images with minimal noise. Any conclusions and insights based upon those feature measurements can be unreliable and biased. As SNR decreases the meaningful signal variations among the different images becomes lost to the noise. As a result, the extracted features begin to characterize the behavior of the noise instead of the signal. Ground truth features should be measured from very high SNR images so feature values are predominantly a function of signal only, and minimally influenced by noise. Differ-

Bajcsy, Peter; Brady, Mary; Chalfoun, Joe; Lund, Steven; Majurski, Michael.
"Methodology for Increasing Image Feature Measurement Accuracy."
Paper presented at Computer Vision for Microscopy Image Analysis (CVMI), Las Vegas, NV. June 27, 2016 - July 1, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

ent types of features might display various robustness to noise. These effects should be investigated before interpreting the features. Figure 1 shows the effect of image noise, displaying a feature value scatterplot for Texture.Average.ENTROPY at an SNR of 2, plotting the measured values against the ground truth values. The red $y = x$ line with a slope of 45 ° denotes where the measured feature value (y-axis) is equal to the ground truth feature value (x-axis). With a correlation value of $\rho = 0.584$ there is only a vague linear relationship between the measured and ground truth values. All features are drawn from Bajcsy *et al*. [1] and their mathematical definitions are available at https://isg.nist.gov/deepzoomweb/stemcellfeatures.

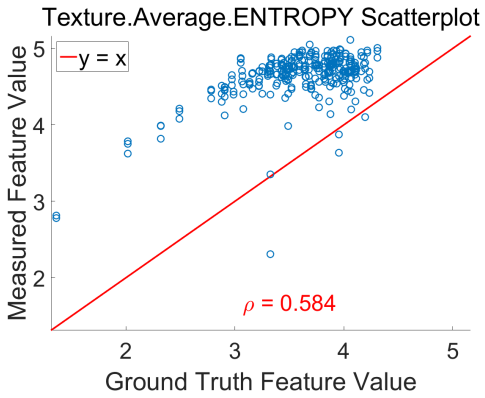


Figure 1. Example low quality (SNR of 2) image feature measurements which do not correlate well with ground truth. The correlation (ρ) between measured and ground truth feature values is 0.584. The red $y = x$ line with a slope of 45 ° denotes where the measured feature value (y-axis) is equal to the ground truth feature value (x-axis).

Previous research has addressed either image quality assessment or the accuracy of feature based application specific results. The intermediate step of analyzing feature quality has been neglected. Image quality assessment utilizes two types of metrics, those based on quantitative deviation from ground truth (SNR, PSNR, MSE [19]) and those designed to mimic human visual perception (FSIM [19], RFSIM [18], MS-SSIM [15], SSIM [14]). In order to evaluate the quality of de-noised images researchers have created synthetic noise models and optimized the selection of de-noising algorithms over using synthetic images [2, 6, 7, 11, 12, 13, 17]. In contrast, we leverage measured images, both high quality ground truth signal and noise. We then create evaluation data based only on measured images. Our methodology of operating on image features is less dependent upon the final application problem.

This paper presents an experimental optimization methodology for improving the measurement accuracy of image features. This methodology consists of four steps: (1) sequester a small subset of the specimen and acquire a set of reference signal images with the highest quality (SNR) possible, (2) acquire a set of background noise images typical of the target experiment, (3) combine the high quality images and the background noise images to create a set of pseudo-real images with an SNR typical of the target experiment, and (4) optimize the application of an ordered set of image processing algorithms in order to maximize the correspondence in feature values measured from the reference signal images and the pseudo-real images.

The low SNR (pseudo-real) images mimic those of the target experiment but contain a known signal component. Therefore features computed from the pseudo-real images can be compared with features computed from the reference signal images. This enables the design, optimization, and refinement of the target experiment image processing pipeline with respect to accurately measuring the image features of interest. Figure 2 shows an overview of the proposed methodology. The reference signal and measured noise images are combined into a pseudo-real image typical of the target experiment. Multiple image processing pipelines are applied to the pseudo-real image generating processed images. Features are then extracted from the processed images and compared to feature values extracted from the reference signal images.

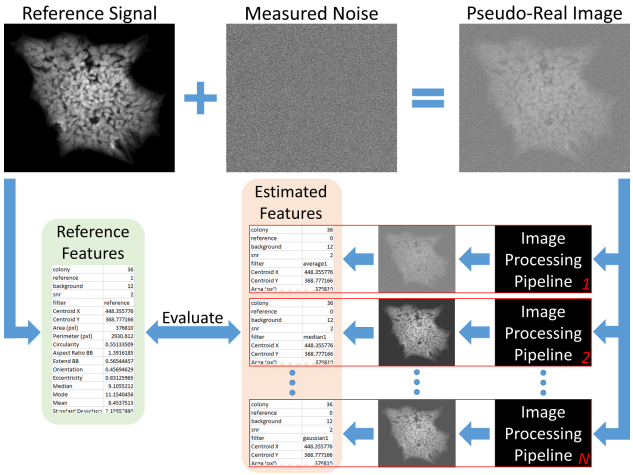


Figure 2. Overview of the methodology for improving the accuracy of measured image features. The reference signal and measured noise images are combined into a pseudo-real image typical of the target experiment. The pseudo-real image is passed through an image processing pipeline to generate a processed image. Features are extracted from the processed image and compared to features extracted from the reference signal image, enabling the evaluation of several processing pipelines to determine which improves the accuracy of the extracted features best.

Pseudo-real images (created by combining measured images) are preferable to synthetic images (created entirely from computer simulation of signal and noise models) because they are more relevant to the imaging experiment in question. Both pseudo-real images and synthetic images

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

contain a known signal component enabling comparison to ground truth. However, the synthetic image noise model might not match the target imaging experiment. By relying on experimentally observed reference signal and noise the pseudo-real images avoid this problem. The downside of the pseudo-real images is that any noise present in the reference signal images cannot be overcome, setting an upper limit on the improvement in feature measurement accuracy. By allowing the experimentalist to define the reference signal and the expected noise the image quality model can be reduced to just SNR, assuming the selection of reference signal images is such that the underlying population of signal profiles is sufficiently narrow that the optimal filter is principally a function of SNR and not the signal itself. In cases where that is not true, the reference image set can be refined until the assumption is met. If required, this methodology can be performed multiple times to optimize filter selection for each sub-type of signal. Other factors affecting filter selection, which might be experiment dependent, will be defined by the experimentalist as part of acquiring the reference signal and noise images.

2. Methods

This experimental methodology for increasing image feature measurement accuracy consists of two major stages. The first stage is the creation of the pseudo-real images and the second stage is the optimization and evaluation of the image processing required to increase the accuracy of the measured image features in generated pseudo-real images.

2.1. Pseudo-Real Image Creation

In order to evaluate the accuracy of extracted image features, a known reference signal similar to ones own experiment is needed. Such a signal is acquired by imaging a representative subset of the experimental specimen with the highest image quality possible. Since this sample is not being relied upon for the actual experiment feature data, it is not subject to the same experimental constraints which limit image quality. For example, a longer exposure time than is practical for the real experiment could be used. These high quality images are called the reference signal because they contain the reference foreground signal from which the ground truth image features are extracted.

The next step is to acquire a sample of the background noise expected in the target experiment. These background images should mimic all noise sources in the target experiment while containing no foreground data. This is done by imaging a background area with the same acquisition setup as used in the actual experiment.

The pseudo-real images are created by combining the reference signal and measured background noise images. Thus the pseudo-real images mimic the expected target experiment images while containing a known reference signal component. They are created by multiplying the reference signal images by a scalar and then adding the background noise. The following procedure is used to create the pseudo-real images with a desired SNR.

1. Compute the mean of the foreground pixels (as identified by segmentation), μ_F, from the reference signal image, I_F.
2. Compute the standard deviation of the background pixels, σ_B, from the background noise image, I_B.
3. Given a desired SNR value of k for the pseudo-real image, compute the rescale factor $(k * \sigma_B)/\mu_F$.
4. Multiply the reference signal image by the rescale factor then add the background images, $I_k = \frac{k*\sigma_B}{\mu_F} \times I_F + I_B$.

This pseudo-real image creation procedure is shown in Figure 3. The formula used to identify the rescale factor for a specified SNR is derived from the Rose criterion as described in [16].

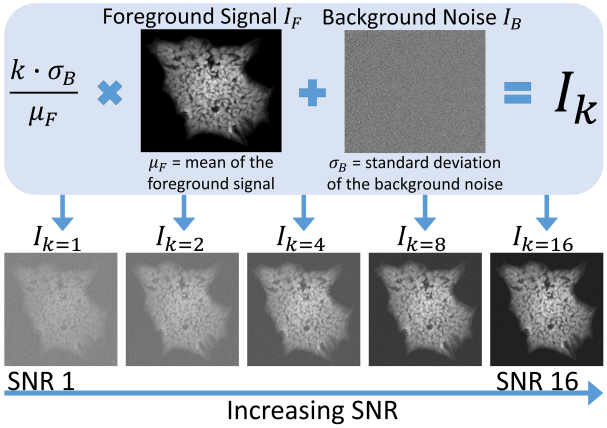


Figure 3. Diagram outlining the pseudo-real image creation. The reference signal (foreground) image is scaled by a factor $(k * \sigma_B)/\mu_F$ and added to the measured background noise image to create a pseudo-real image with a specified SNR value. Note: all displayed images were auto-contrasted using the same algorithm for visual clarity.

2.2. Process Optimization

We then determine an ordered set of image processing operations required to improve the feature measurement accuracy. Ground truth feature values are measured from the reference signal images, denoted *Reference*. These are then compared with features measured from the pseudo-real images after processing, denoted *Processed*. Correlation (ρ), described by Eq. (1), is used to compare sets of feature values.

$$\rho = \frac{cov(Processed, Reference)}{sd(Processed) * sd(Reference)} \quad (1)$$

The correlation coefficient measures the linear relationship strength between the processed feature values and their corresponding reference values. A correlation coefficient of 1 indicates that there are scalar values $a > 0$ and b such that the equation $Reference = a*Processed+b$ holds exactly. When primary interest lies in the relative sizes of pairwise distances among any set of points, as opposed to the exact values of the features themselves, correlations near 1 can be interpreted as meaning the two quantities are nearly equivalent.

3. Experimental Results

This section presents experimental results obtained by applying the proposed optimization methodology to cellular microscopy images. There are many factors involved in keeping cells alive, but with respect to imaging, prevention of photo-toxicity is the important one. It is desirable to have the highest SNR images possible to discern biological information with the highest fidelity. However, acquiring these high SNR images requires exciting the fluorophores within the cells with high-intensity light. This exposure damages or kills the cells (due to photo-toxicity) and causes accelerated photo-bleaching of the specimen. Note that we are mainly concerned with fluorescent imaging modalities where the sample must be probed with excitation light. For transmitted light modalities photo-toxicity is less of a concern. Thus, our primary interest lies in strategies to balance the competing interests of using minimally invasive imaging techniques to avoid affecting cell behavior or survival and acquiring high quality images which contain the required information.

3.1. Measured Microscopy Images of Cells

The target imaging experiment consists of a time-lapse acquisition of the H9 human embryonic stem cell (hESC) line over the course of 5 days on a microscope equipped with a controlled environment incubation chamber (Kairos Instruments LLC, Pittsburgh, PA). This cell line was engineered to produce green fluorescent protein (GFP) under the influence of the native OCT-4 promoter using a published homologous recombination plasmid construct developed by the James Thomson lab [20] and obtained from Addgene (Addgene, Cambridge, MA). Experimental imaging is performed using a Zeiss 200M microscope (Carl Zeiss Microscopy, LLC, Thronwood, NY) every 45 minutes via a Coolsnap HQ camera (Photometrics, Tucson, AZ) in a grid of 16×22 field of views (FOVs) with 10% overlap covering approximately 180 mm^2. Each individual FOV (image) is 1040×1392 pixels.

The individual target experiment images are stitched into a single mosaic per time point using MIST (Microscopy Image Stitching Tool) [3]. Foreground and background masks are generated by segmenting the phase-contrast stitched images using the Empirical Gradient Threshold technique [5]. The stitched mosaic images are flat-field corrected and background subtracted [4]. Using the foreground masks a set of 61 intensity and texture image features, taken from [1], are extracted from each colony. The intensity features are statistical moments: mode, mean, mode, standard deviation, skewness, kurtosis, etc. The texture features are based on Haralick texture features [10] which generate four values per feature type, the average amplitude, principle component angle, orthogonal component angle, and principle component value.

Since this is a time-lapse experiment, the cells need to be kept alive and minimally disturbed by the high intensity light used in imaging. Therefore, experimental conditions constrain imaging to phase contrast (less-damaging transmitted light) and low SNR fluorescent imaging. Many regions of interest exhibit SNRs of roughly 2. The goal of the cell imaging is to classify stem cell colonies based on homogeneity and to analyze the homogeneity distribution of these colonies through time. The classification of cell colonies is based on the intensity and texture features extracted from the fluorescent images. Therefore, it is important to compute the features with the highest accuracy possible under these circumstances. We apply the proposed methodology on this problem to find the optimal image processing steps that increase the accuracy of the measured features.

3.2. Pseudo-Real Image Creation

For this application the reference signal image dataset consists of 100 stem cell colonies imaged in the fluorescent channel with a long exposure time and high power excitation light to create very high SNR images. All of these colonies fit within a single FOV and are larger than 1000 pixels in area. It is important to note that in acquiring these images with the aforementioned acquisition parameters, the colonies were both damaged and photo-bleached, making this acquisition method unsuitable for the target time-lapse experiment.

Typical background noise for the target experiment is acquired by imaging the specimen background consisting of cell culture media, culture dish, and any extracellular matrix protein coatings under the same acquisition parameters as the real experiment. In addition to any background autofluorescence, the CCD camera noise is captured in these background images. We acquired 30 background images with different spatial locations on the plate typical of the conditions expected in the target time-lapse imaging experiment.

Conditional random sampling is applied to the set of

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

100 reference signal colony images and 30 measured background noise images to produce the set of pseudo-real images. Each colony image is combined with 3 background images. Each background image is selected 10 times for a total of 300 combinations. The subsampling, as opposed to a complete factorial design, is used to restrict computational requirements to a reasonable level. Next, each colony image containing ideally pure signal is combined with its selected backgrounds to create 5 target SNR levels (1, 2, 4, 8, and 16). The colony images were segmented using a manually selected threshold (foreground is greater than 500 intensity units) to set the background of the image to 0. Before this adjustment the reference signal image background (non-colony pixels) contained just dark current noise from the CCD camera with intensity values of approximately 200 units. The colony foreground contains pixels of approximately 4000-8000 grayscale intensity units coming from a 14bit CCD camera with an output range of 0-16284 intensity units.

3.3. Optimization of Image Processing Filters

For this application we are interested in selecting the spatial image processing filter and kernel size for each feature which produces the most accurate measurement of that feature. While this methodology enables the design and optimization of arbitrary image processing pipelines with respect to feature measurement accuracy, we have limited the complexity of the processing pipeline to a depth of one operation and a small set of manually selected spatial image filters (Average, Median, or Gaussian) [8, 9]. These filters were chosen because they are commonly used methods of reducing image noise. Each filter is parameterized by a kernel size of which six were tested (3x3, 5x5, 7x7, 9x9, 13x13, 17x17). In order to evaluate the image processing, each feature was computed for each combination of filter type and kernel size.

3.4. Numerical Results

Each filter and kernel size combination is applied to the pseudo-real images and all 61 features are extracted from the processed images. This enables the analysis of how the feature values change as a function of the image filter, kernel size, and image SNR. The target experiment of this study has an expected image SNR of 2. The optimal filter and kernel size can be selected for each feature by selecting the filter and kernel which maximizes the correlation in Eq. (1) between the processed and reference feature values.

Applying the filter selected for each feature increases the average correlation from 0.601 to 0.919. Of the 61 features evaluated, 77% are optimized with the Gaussian filter, 18% with the Average filter, 3.3% with the Median filter, and 1.6% with No Filter. Kernel sizes 5x5 and 7x7 are the most common at 20% and 61% respectively. The majority of the features (57%) have the same optimal filter and kernel size, 7x7 Gaussian. Figure 4 shows a histogram of the feature correlation values for no filter and the optimal filter per feature, highlighting the increase in correlation.

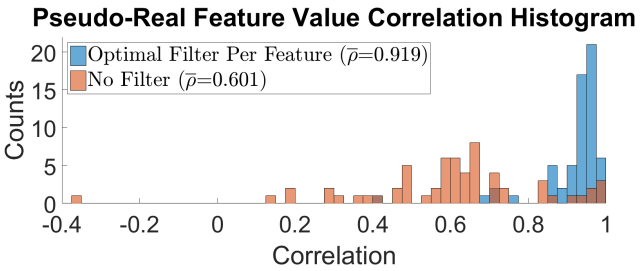


Figure 4. Histogram of the extracted feature correlation with ground truth values for pseudo-real images with an SNR of 2. Average correlation ($\overline{\rho}$) is listed in the legend.

The feature correlation (Figure 4) without filtering has an average correlation of 0.601 and only a few features with correlations above 0.8. Once filtering has been performed the majority of the features have correlations with ground truth above 0.8. There are two groups of features that do not respond well to filtering. The first group contains just the statistical moment feature Mode ($\rho \approx 0.4$). The second group contains all of the Haralick principle component angle texture features. All optimal filter per feature correlation values below 0.9 are principle component angle texture features with the exception of Mode. Without these two groups the optimal filter per feature average correlation is 0.942.

By averaging correlation across all features a single optimal image processing filter, the 7x7 Gaussian, can be found for this experiment. Doing this results in a slight loss in average accuracy compared to selecting the optimal filter for each feature. Among the features whose per feature optimal filter differs from 7x7 Gaussian there is a 1.95% loss in average correlation and a 10.04% increase in average feature residual RMSE Eq. (2).

$$residualRMSE = \sqrt{\frac{\sum_{i=1}^{N}(Proc_i - Ref_i)^2}{N}} \qquad (2)$$

The correlation metric selects the filter which results in the strongest linear relationship between the ground truth feature values and the processed feature values. If exact feature values are required a linear transformation can be applied to the processed feature values. This is demonstrated in Figure 5 where the best filter for the feature Texture.Average.ENTROPY at an SNR of 2, 5x5 Gaussian, results in a bias in the processed feature values. This bias is corrected with a linear transformation (slope $a = 1.143$ and intercept $b = -1.115$) reducing the residual RMSE from 0.545 to 0.135.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

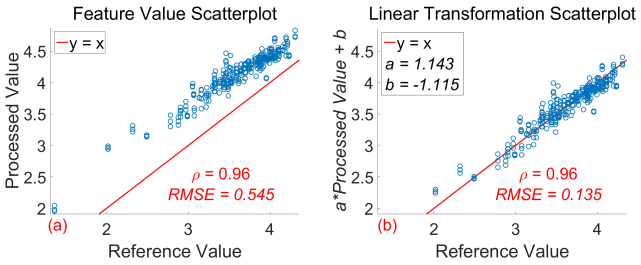


Figure 5. Feature Texture.Average.ENTROPY (SNR of 2) processed with the 5x5 Gaussian filter. The original processed feature values are shown in (a) with a bias residual RMSE of 0.545. The linear transformation of the processed feature values is shown in (b) with a lower residual RMSE value of 0.135.

To examine the relationships between the processed image feature values and the ground truth feature values a series of exploratory plots were generated. The first, shown in Figure 6, contains scatterplots of the processed feature values plotted against the ground truth feature values. This figure is organized into a two dimensional grid of scatterplots. Within each plot the feature value for an individual image is shown as a single point and the line marks $y = x$, where the processed value equals the reference value. The text superimposed on each scatterplot is the corresponding correlation coefficient, see Eq. (1).

With no filter applied (indicated by the 1x1 kernel size) there is a clear bias in the computed features that decreases with processing. As the kernel size increases the correlation values increase and the feature values show a reduction in bias. The effects of different image filter types is most evident in the 3x3 kernel size plots. Of the 3x3 filters, the Average filter has th least bias and highest correlation. Moving across the row of 3x3 kernel size scatterplots, the correlation decreases and the distribution gets further from the $y = x$ line. Increasing the kernel size reduces the disparity in results between filter types. The optimal filter for this feature is Gaussian with kernel size 5x5.

Due to the high dimensionality of the feature accuracy data it is hard to conceptualize the full picture. Therefore, a summary plot was created where the correlation values previously printed on the scatterplot are plotted as a function of image feature, filter type, kernel size, and image SNR. This plot is shown in Figure 7 and can be found in supplementary document 1. Each image SNR block contains 4 sub-blocks, the Gaussian filter block 'Gau', the Median filter block 'Med', the Average filter block 'Ave', and the No filter block 'None'. Within each filter block, the kernel size increases from bottom to top, 3 to 17. Per column within each SNR block the maximum correlation value is shown by printing the relevant kernel size. Figure 7 shows that there is considerable variability in the optimal filter and kernel size between different features. Overall, as the image SNR increases the optimal kernel sizes shrink.

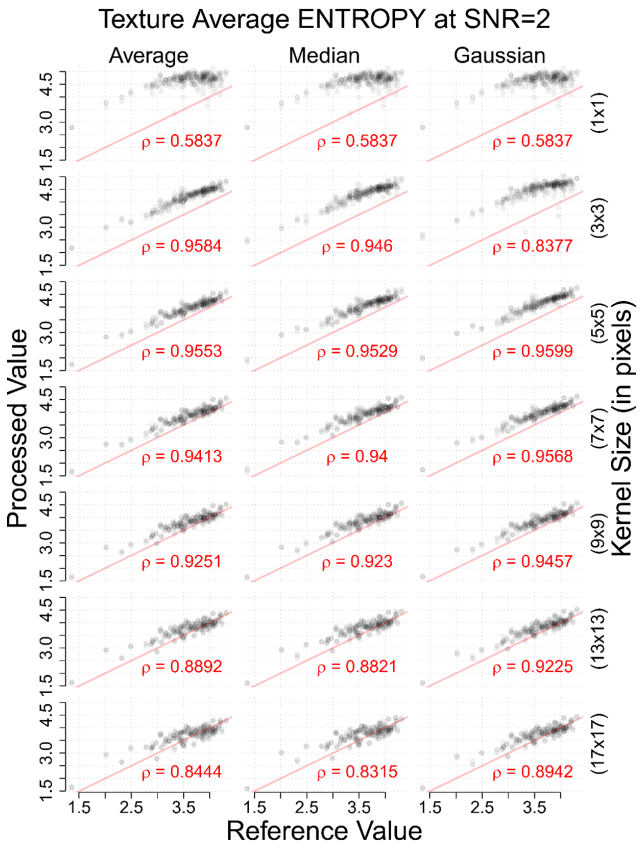


Figure 6. Feature value scatterplots for the feature Texture.Average.Entropy at an SNR of 2 given the different filter types and kernel sizes. This figure is organized into a two dimensional grid of plots. Within each plot the feature value for an individual image is shown as a single point. The line marks $y = x$, where the processed feature value equals the reference feature value. The superimposed text on each scatterplot is the correlation coefficient (ρ) for that scatterplot.

Reducing the dimensionality of the data once more is done by averaging correlation values across all features to produce a single value per filter type, image SNR, and kernel size. This creates plots where feature correlation is shown as a function of kernel size for each image SNR and filter type. Figure 8 depicts plots of these feature correlations processed with Average, Median, and Gaussian filters, where each point shows correlation averaged across all 61 features.

4. Discussion

There are several general observations that can be gleaned from Figure 8. First, a dominant factor in determining the processed feature measurement accuracy is the image SNR. The higher the acquired image SNR the more accurate the feature measurement which is logical since higher SNRs have less noise to distort the feature measure-

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

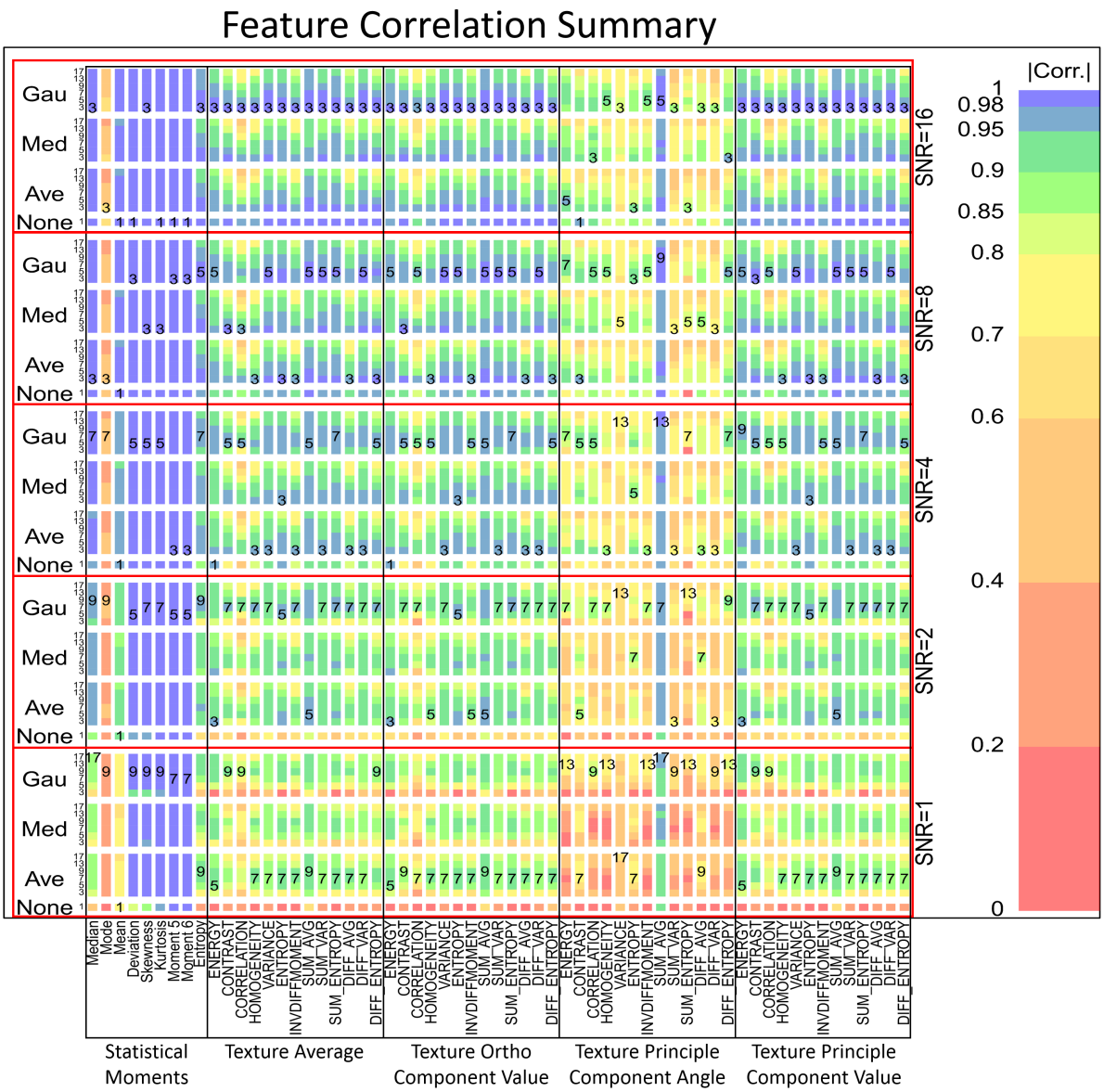


Figure 7. Correlation summary plot. Each image SNR block contains 4 sub-blocks, the Gaussian filter block 'Gau', the Median filter block 'Med', the Average filter block 'Ave', and the No filter block 'None'. Within each filter block, the kernel size increases from bottom to top, 3 to 17. Per column within each SNR block the maximum correlation value is shown by printing the relevant kernel size.

ments. If the image SNR is high enough there is little to no accuracy gained by filtering the images. For example, at an SNR of 16 a 3x3 kernel provides a minor increase in feature measurement accuracy, but a 5x5 kernel provides equal or worse accuracy than no filter. Second, as the image SNR decreases, larger filter kernels are required to obtain a given level of feature measurement accuracy. For example, at an SNR of 4 a 3x3 Gaussian kernel produces roughly the same accuracy as a 5x5 Gaussian kernel at an SNR of 2. Third, Gaussian filters require a larger kernel size to accomplish the same effect as the Median or Average filters.

The time-lapse stem cell colony imaging experiment presented here has an SNR of approximately 2. Given that constraint, the optimal image filter and kernel size for each feature should be selected such that the correlation between the processed feature values and ground truth feature values is maximized. The per feature filter selection accuracy data is available in supplementary document 2 for each image SNR level. Looking at just the filter type selection for an SNR of 2, 77% are optimized with the Gaussian filter, 18% with the Average filter, 3.3% with the Median filter, and 1.6% with No Filter. The optimal kernel size distribution is

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

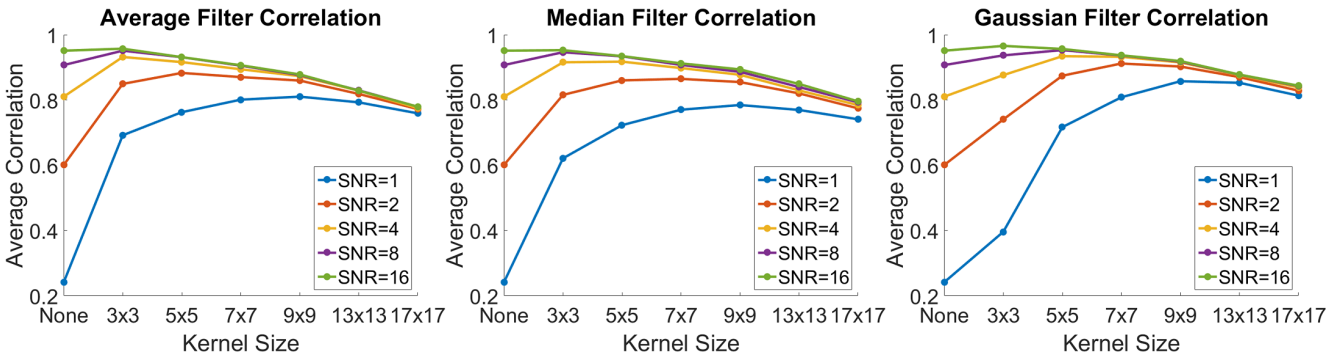


Figure 8. Average feature correlation values for each Filter, SNR, and Kernel size combination. The first plot was processed with an Average filter, the second a Median filter, and the third a Gaussian filter. Within each plot correlation is shown as a function of kernel size for multiple SNR values.

more spread out with 60.65% being 7x7, 19.67% 5x5, 8.2% 3x3, 6.56% 9x9, 3.23% 13x13, 1.64% No Filter, and 0.0% 17x17. The most common filter and kernel size combination is 7x7 Gaussian which is optimal for the majority of the features (57%). This effect shows up in Figure 7 as a fairly consistent row of '7's written within the 'Gau' block of 'SNR=2'.

Texture features which compute a principle component directionality angle did not improve nearly as much as the other evaluated features. These features accounted for all but 1 of the features that did not obtain a correlation of 0.9 or greater under any considered filter. This shows up in Figure 7 as a vertical block of lower correlation values across all SNRs.

Whether one selects a single image processing filter for the entire experiment or a filter per feature, these results are only relevant for the target experiment under consideration. The numerical results cannot be generalized but the methodology can. Changes in the target experiment (different cell line, different features, etc.) would require this pre-experiment to be redone in order to find the optimal filter(s) and kernel size(s) for the new target experiment. The power of this approach is its flexibility and extensibility. This optimization methodology can be applied to different experiments, image conditions, image modalities, and image features. For small scale experiments it might not be reasonable to perform such a pre-experiment to help design the target experiment and its data processing. However, as long as the pre-experiment does not constitute an unreasonable effort, it can help inform the accuracy of the target experiment.

5. Conclusions

This work was motivated by a desire to understand the impact on stem cell colony classification when using image features derived from low SNR images. We devised a methodology to quantify the improvement of feature measurement for a given image pre-processing method. As a proof of concept, we chose three basic filtering techniques as pre-processing steps. We found that selecting the best filter per feature produces a 53% improvement in feature correlation with ground truth, from 0.6 to 0.92. Selecting a single filter for all features results in a 1.95% reduction in correlation and a 10% increase in residual RMSE.

6. Future Work

We intend to measure the impact of using image features derived from pre-processed images on colony classification accuracy. The pool of image processing operations is going to be expanded to include more advanced image enhancement and noise reduction algorithms.

7. Acknowledgments

This work has been supported by NIST. We would like to acknowledge the team members of the computational science in biological metrology project at NIST for providing invaluable inputs to our work. We would also like to thank specifically Kiran Bhadriraju, Greg Cooksey, Michael Halter, John Elliot, and Anne Plant from Biosystems and Biomaterials Division at NIST for acquiring the image datasets.

8. Disclaimer

Commercial products are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

References

[1] P. Bajcsy, A. Vandecreme, J. Amelot, P. Nguyen, J. Chalfoun, and M. Brady. Terabyte-Sized Image Computations on

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Hadoop Cluster Platforms. *2013 IEEE International Conference on Big Data*, pages 729–737, oct 2013.

[2] S. Bharadwaj, H. Bhatt, M. Vatsa, R. Singh, and A. Noore. Quality Assessment Based Denoising to Improve Face Recognition Performance. In *Computer Vision and Pattern Recongnition Workshops (CVPRW)*, pages 169–174, 2011.

[3] T. Blattner, J. Chalfoun, B. Stivalet, and M. Brady. A Hybrid CPU-GPU System for Stitching of Large Scale Optical Microscopy Images. *International Conference on Parallel Processing (ICPP)*, 2014.

[4] J. Chalfoun, M. Majurski, K. Bhadriraju, S. Lund, P. Bajcsy, and M. Brady. Background Intensity Correction for Terabyte-Sized Time-Lapse Images. *Journal of Microscopy*, 257(3):226–238, 2015.

[5] J. Chalfoun, M. Majurski, A. Peskin, C. Breen, P. Bajcsy, and M. Brady. Empirical Gradient Threshold Technique for Automated Segmentation Across Image Modalities and Cell Lines. *Journal of Microscopy*, 260(1):86–99, 2015.

[6] M. Elad and M. Aharon. Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. *IEEE Transactions on Image Processing*, 15(12), 2006.

[7] R. Eslami and H. Radha. Translation-Invariant Contourlet Transform and its Application to Image Denoising. *IEEE Transactions on Image Processing*, 15(11):3362–3374, 2006.

[8] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice-Hall, New Jersey, 2nd edition, 2008.

[9] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using Matlab*. Pearson Prentice Hall, New Jersey, 2004.

[10] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 1973.

[11] B. Matalon, M. Elad, and M. Zibulevsky. Improved Denoising of Images Using Modelling of a Redundant Contourlet Transform. *Optics & Photonics 2005*, 2005.

[12] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain. *IEEE Trans Image Processing*, 12(11):1338–1351, 2003.

[13] J.-L. Starck, E. J. Candès, and D. L. Donoho. The Curvelet Transform for Image Denoising. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 11(6):670–84, 2002.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[15] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multi-Scale Structural Similarity for Image Quality Assessment. *IEEE Asilomar Conference on Signals, Systems and Computers*, 2, 2003.

[16] R. Watts, Y. Wang, P. A. Winchester, N. Khilnani, and L. Yu. Rose Model in MRI: Noise Limitation on Spatial Resolution and Implications for Contrast Enhanced MR Angiography. In *Intl. Society Mag. Reson. Med.*, volume 4, page (8) 462, 2000.

[17] K. Youssef, N. N. Jarenwattananon, and L. S. Bouchard. Feature-Preserving Noise Removal. *IEEE Transactions on Medical Imaging*, 34(9):1822–1829, 2015.

[18] L. Zhang, L. Zhang, and X. Mou. RFSIM: A Feature Based Image Quality Assessment Metric Using Riesz Transforms. In *International Conference on Image Processing (ICIP)*, pages 321–324, 2010.

[19] L. Zhang, L. Zhang, X. Mou, and D. Zhang. FSIM: A Fast Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(8), 2011.

[20] J. A. Zwaka, Thomas P and Thomson. Homologous Recombination in Human Embryonic Stem Cells. *Nature biotechnology*, 21(3):319–321, 2003.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Event model to facilitate data sharing among services

Closing the Gap between Research and Implementation

O. Kotevska, A. Lbath*, J. Gelernter
National Institute of Standards and Technology
University of Grenoble Alpes, France*
{olivera.kotevska, gelern} @nist.gov, ahmed.lbath@imag.fr*

***Abstract*—Development of smart city services is presently hindered because the data is too heterogeneous, despite the increasing availability of data in open data initiatives. We determined metadata fields and semantics for our proposed model after a survey of other event models, and after its trial implementation in actual applications. Our event metadata model is unique among event models in being extensible as needed. We illustrate our model by showing it with different data types, and we validate it using real-world data in a prototype smart cities service for pedestrian safety. Wide adoption of our event metadata model has the potential to broaden the number and scope of smart city services.**

***Keywords*—Complex Event Processing, CEP, event metadata model, event model, smart city service**

I. Introduction

Problem. Half of the world population lived in cities by 2013. Historically, people live in and work for their cities; we want to make the cities work for them. Or rather, to make their devices work for them by creating city service applications. Complex event processing applications correlate data streams in real time as events occur, using pre-defined rules to identify events of interest. A common data structure will allow different data streams to be used by multiple services more easily.

Others' approaches. Event metadata models for Complex Event Processing provide the time and data backbone to allow those data streams to work together. Event models tend to be rich in metadata fields which may or may not be relevant to the service at hand. The impracticality of the metadata-rich models is not recognized because use cases are rare. Also, when data is provided, it is often simulated to match a model – which does not demonstrate whether or not a model is robust for a particular data stream.

Our approach. Our event model defines only necessary fields, and gives parameter wildcards for potential metadata expansion in three categories (*EventProfile* for the data stream, *EventSource* for the device that provides the data stream, and *Event*). These parameters are added based on the service by the developer at the time of model implementation. Our model is validated in a smart cities case study.

Significance. Developers' adoption of a standard event model – that is, a package that any data could fit into for delivery into applications in the upper layer of the Internet – could advance smart city services. Different services might access the same data concurrently, so the model would facilitate data sharing as well as integration.

Case study in city safety. We created an algorithm for pedestrian safety using real-world data, and then supplied the algorithm with data from our event model to test our model's usefulness.

We devised the case study in answer to an on-going need by local government to improve safety in Montgomery County, Maryland, U.S.A. In 2007, a Pedestrian Safety Initiative was introduced in Montgomery County that used education, streetscape and road signs, as well as traffic calming and road rule enforcement to lower the number of accidents.[1] Our application is intended for county representatives, and adds to Pedestrian Safety Initiative by showing where the most accidents are predicted to occur. The county can then better deploy resources to change street signs or deploy traffic police, for example, to heighten safety at the times and locations predicted to be unsafe.

This Introduction is followed with Section 2 on Background and Related work. Section 3 presents our event model; in Section 4 we demonstrate evaluation scenario; in Section 5 we discuss about our results and Section 6 summarizes the conclusions and future work.

Our contributions are (1) a generic, highly-scalable semantic model as a core component of complex event processing architecture, (2) validation of the proposed event model in an algorithm for a smart cities problem of pedestrian safety, and (3) bringing to the attention of the research community an Internet of Things testbed that includes both sensor and human data.[2]

[1] http://www.montgomerycountymd.gov/DOT-PedSafety/overview.html; https://volunteer.truist.com/mcvc/org/opp/10610402110.html

[2] https://data.montgomerycountymd.gov/, with data updated regularly

Government work not protected by U.S. copyright.

Gelernter, Judith; Kotevska, Olivera.
"Event model to facilitate data sharing among services."
Paper presented at 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA. December 12, 2016 - December 14, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

II. Background and Related Work

A. Complex Event Processing (CEP)

Complex event processing is a strategy to create applications, or *services* in which data streams containing events produced by *resources* such as people, devices or sensors are filtered for particular event changes of interest, called *instances*. These changes then automatically trigger some response. Events are given names so that they can be shared among services. To enable sharing, common metadata and sometimes also vocabulary for the names comes from *ontologies*.

The diagram below shows event data in basic Complex Event Processing. Data is received from multiple streams, events are detected, and an appropriate response is triggered. This sequence diagram the Figure resembles the standard event driven architecture proposed by Fujitsu [11], Microsoft [5], IBM [20] and Oracle [15].

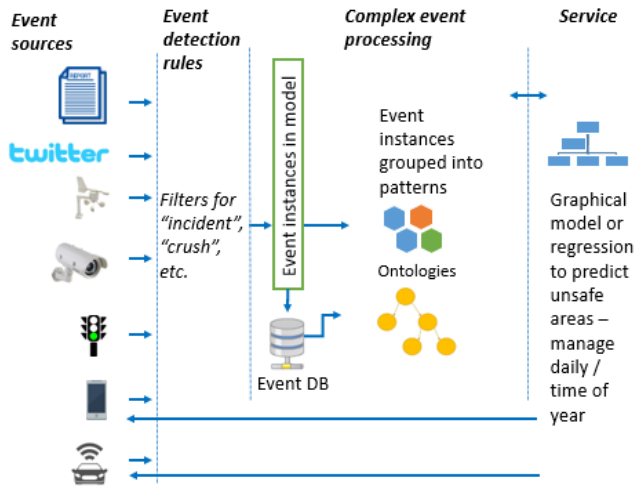


Fig. 1. Complex event processing architecture, with arrows showing event data sequence direction. An event model is a component.

Note that the vertical box in Fig. 1, "Event instances in model," where the data is organized to be stored in database server or event cloud, is central to the process.

Queries to the event processing system might be in SPARQL for RDF, TESLA–a formally defined Event Specification Language [6], or EP-SPARQL [2], or in Event Processing Language,[3] or Big Data Processing Language (BDPL) [24].

These rules have the ability not only to indicate events relevant to the application, but also when data is faulty or noisy or missing information, or includes non-events [1]. Response to data uncertainty can make the system more robust. For example, events missing values can be marked incomplete, and system maintenance would analyze marked records to determine what had gone wrong.

[3] https://docs.oracle.com/cd/E13157_01/wlevs/docs30/epl_guide/overview.html

B. Alternatives to this architecture

The event model is one part of the CEP architecture. Other proposed middleware solutions ensure that low level signals can work together without use of an event model. For example, the IoT Semantic Smart Gateway Framework [14], and the Smart M3 open source software platform that provides infrastructure between resources [23]. Even these solutions require common terms so that different devices can communicate – in what makes the solutions "smart".

C. Event model in Complex Event Processing

An event model is core to Complex Event Processing architecture, so it is understandable that more than a dozen event models have appeared in the academic literature since 2000. Some are specific to data type or domain type, or must be paired with certain controlled vocabularies in ontologies, or with particular query languages.

Event models specify metadata fields for the data streams that carry or pertain to events. The most basic metadata fields for event models are data type, and time. The idea of the model primitives is that the basic data message is independent of any application. What metadata fields go beyond the primitive? Authors from [17] explain that data fields such be considered during the "thought process" of setting up a system. This is the logic we follow when specifying "Parameters" for our model. Other researchers [19] assume that more primitive are better. See the comparison chart with evaluation metrics in the Appendix.

D. Role of the ontologies in event models

Ontologies are required in event models such as Event F [19][19], OntoEvent [16], REseT [25], and LODE [21]. In some cases, as in OntoEvent and REseT, the ontology was created specifically for the event model.

The metadata fields in our model, for the most part, align with those in the DOLCE ontology (See Fig.1) That will make it easier for data in different streams to fit into the same event model, and it will help insure interoperability. These event models are called semantic because the meaning of the data field titles is essential to the model.

Requiring an ontology on a particular topic for an event model can restrict data types that can be used, as well as the domain. Authors in [13] have a method to develop an ontology for traffic violations. In our particular application, the World Wide Web Consortium is developing a Traffic Event Ontology, which we could use for at least one data stream, but this ontology is not available as of the writing of this paper.[4] A domain-specific ontology might be useful to extract events from a natural language data stream, but it is probable that Named Entity Recognition tools such as the Stanford NER

[4] https://www.w3.org/community/traffic/

toolkit would do well in extracting proper nouns and locations required.[5]

E. How event models have been validated

Many event models in the academic literature have not been validated; the models are either entirely theoretical, or use data that has been created to fit the model. EventShop is an event processing platform that can be used to validate components of the architecture. It accepts data streams and includes modules to query the data that will isolate event instances relevant to the application. Their events are put into a grid structure called E-mage, that can integrate events from heterogeneous data streams [17].

By contrast, to validate our event model, we have devised a service that incorporates multiple data streams. We have vetted our event model when real-world data streams fit the model and then are accepted into the service.

III. Our Event Model

Our model shown in Fig. 2 below in Unified Modeling Language (UML) has three classes: EventProfile, EventSource, and Event. Location metadata is required but is not in the diagram because location metadata changes classes depending upon the service. For stationary devices, "Location" is stored with the device information, whereas for mobile devices, location will be needed continually to correspond to events. Metadata fields within the three classes of our event model are defined below.

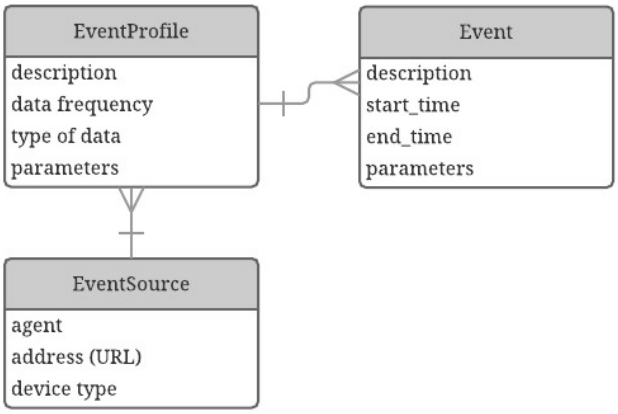


Fig. 2. Our model for event data shown in Universal Modeling Language (UML). EventProfile and EventSource parameters could include data rate for system failure checking, and parameters could include location for devices that are mobile; otherwise, location would be a parameter for Event.

Our model is extensible. Some services will require only its specified metadata fields, whereas other services will require consideration of which parameters are relevant. *EventProfile* and *EventSource* send their data at system start or re-start, whereas the Event class sends data continually. Data from the Event class is filtered using rules or machine learning into instances relevant to the service.

The *EventProfile* class holds metadata about the data stream. "Description" explains what data is being stored for services. "Data frequency" presents the rate data streams are expected to be received, while "type of data" provides information about the data that has been collected. "Parameters" represent any additional information related to the data stream that are important, like measurement type, expected states, or frequency.

The *EventSourc*e class holds the data related to devices or sensors. It has the following attributes:" Agent" provides information for the event source, like county police or weather channel. "Address" is the address from where data was collected or original source of observed data. "Device type" is the type of device, such as temperature sensor, mobile device etc. "Parameters" represents additional information related to the device, such as serial number, model, and battery expiration date.

The *Event* class holds the data related to events in the data stream, like temperature, heart rate, pedestrian movement, or car accident. It has the following attributes: "Description" is auto-filtered from an event text stream or is set up beforehand to describe the events. "StartTime" is the date and time when the event started, "EndTime" is the date and time when event stopped. The frequency of "StartTime" and "EndTime" can be set by adding a parameter to the *EventProfile*. Parameters for the event represent additional information related to the event that are useful for a particular service. The events could be sent in as received, or they could be aggregated and then sent at some pre-determined frequency.

*Ontology***.** The system designer should choose an ontology used by the other applications for wide data sharing among applications. Rather than share or link ontologies [12], an upper level ontology such as DOLCE could be used to bridge domain ontologies.[6] DOLCE includes reasoning without controlled vocabulary, and it was used in the F event model [19]. In aligning our model with DOLCE, Feature is the category for our Description, and Physical Quality is the category for Type.

Uniqueness and benefits. We conducted an extensive survey of event models and found that our event model differs from others in two main aspects: (1) fewer required metadata fields, and (2) some of the metadata is sent once only. Fewer metadata fields allows the model to be maximally re-usable for different data and domain types, although it means that some thought will be needed each time the model is implemented. Sending some metadata once only lessens the data load on the network somewhat to speed processing.

Instances in the event stream. Recall that instances or occurrences are the events being monitored for. A script apart from the model filters the event data stream to recognize instances relevant to the application, and to set bounds on what belongs in the event model. For example, an expected range of values for some metadata field could be specified, and data that

[5] http://nlp.stanford.edu/ner/

[6] http://www.loa.istc.cnr.it/old/q.html

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

went beyond that range would be dismissed. Filtering for the model also could be handled on a data stream-by-stream basis so that a data stream that did not flow at an expected frequency, might indicate sensor or device malfunction.

IV. Case Study

This experiment shows how different forms of event data and metadata in the model are suitable for a smart cities application.

A. Objective

We want to use our event model to supply data for an actual application. Our research question is: *Is it possible to show the influence of time, weather, and community happenings on pedestrian safety incidents in a given location by postal zip code?*

B. Use case and data for the experiment

Events for 2015 traffic incidents, weather, and community happenings for Montgomery County, Maryland are posted on the open web.[7] The county updates the data regularly. We intend to supplement the Pedestrian Safety Initiative in Montgomery County, Maryland, by offering an algorithm to predict unsafe event areas according to zip code. To show differences in time-of-day, the service could output predicted locations every few hours.

C. Method

Event processing methods work well for passive data collection because they allow large quantities of data to be processed if necessary in different locations (also called distributed systems). Event processing also is scalable, and using corrective rules can help make the system fault tolerant. For example, only accept into the model if the time is between 1pm and 6pm Eastern Time.

Getting data into the model. Our data included road violations, weather, and community happenings. Of these, the road violations and community happenings include a lot of text. But event recognition in text data can be difficult due to human language ambiguity. To recognize which events were relevant instances, we experimented by comparing a non-domain specific ontology, the Freebase knowledge-base, and a classifier we trained on a portion of the data. We found that the classifier was the most accurate filter to find relevant instances.

Handling data uncertainty. Uncertainty in the general case could be minimized with the combination of rules (for instance, only accept into the model if it includes a beginning time). Specifically, in our application, we lacked zip codes for some of the community happenings that were city-wide (such as elections), so for these, we entered zip codes for the entire city. Also for the case of missing data in the location fields, we used alternative data entry, like longitude and latitude and get the zip code and city information.

[7] https://data.montgomerycountymd.gov/

Example of data in the model. The following Figures 3,4,5 show how community happenings, weather and traffic violation metadata fit into our event model. We took this data from the Montgomery County open data website for this case study, but in a real time application, the *EventSource* would provide the actual source of the data flow with its parameters.

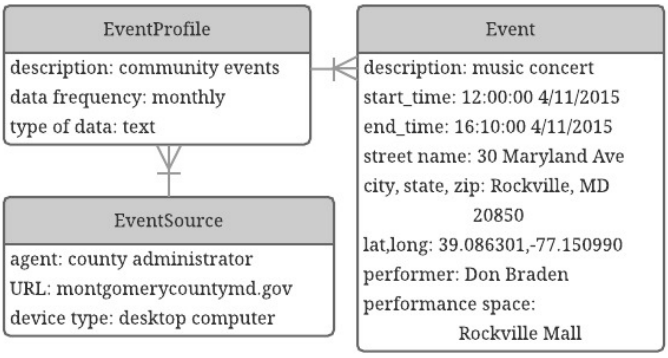


Fig. 3. Community happening metadata in our event model

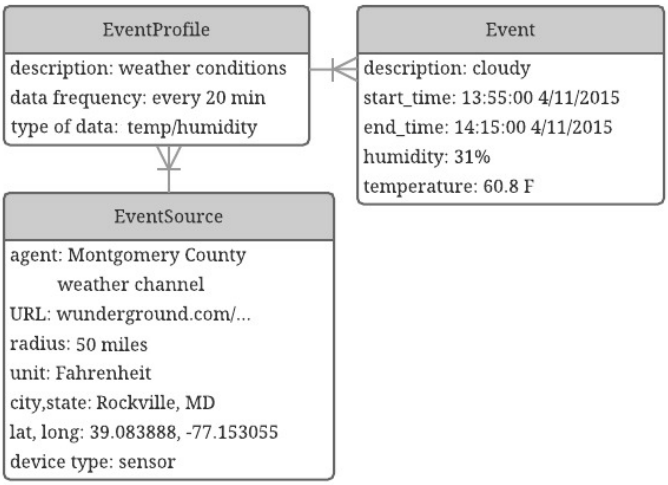


Fig. 4. Weather condition metadata in our event model

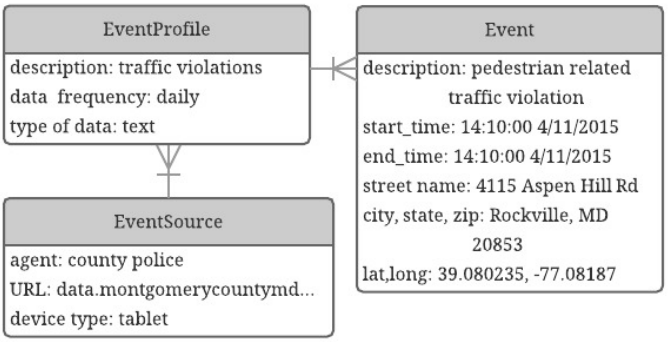


Fig. 5. Traffic violation metadata in our event model

Our metadata fields correspond to categories in the DOLCE ontology. DOLCE allows natural language descriptions, so we leave our text data streams as sent.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

D. Data analytics and findings

The proposed method predicts which county region by zip code will be safest based on the past number of pedestrian incidents per zone. Analysis is performed using the R software environment for statistical computing and graphics. We experimented with both Poisson Regression and Probabilistic Graphical models (PGM) to determine which was more suitable for prediction with our data. Poisson regression is used for point data, such as we have, and it is commonly used for predicting the probability that an event will occur based on several predictor variables that may either be numerical or categorical. PGM has the ability to represent dependencies among events on a graph. The table shows that the statistical method selected, whether a PGM or the Poisson Regression, is not a significant factor in predictive accuracy.

TABLE I. ACCURACY OF TWO METHODS TO PREDICT UNSAFE LOCATIONS

	Probability of a (negative) pedestrian event in a location	
	Probabilistic Graphical Model	***Poisson Regression***
Average accuracy	0.864	0.869

E. Evaluation and visualization

To show the relative effectiveness of the pedestrian safety algorithm on data from the event model, two means of evaluation are used: comparative maps (Fig.6 and 7), and the confusion matrix as performance measures for accuracy calculated in Table 1. Figure 6 show maps for 2015 actual pedestrian incidents for the period from January to March and incidents predicted for the same period using Probabilistic Graphical Models.

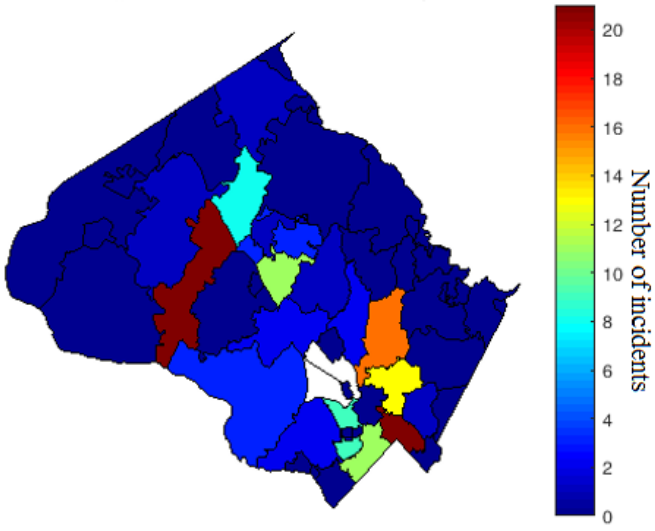


Fig. 6. Actual traffic violations in Montgomery County, Maryland related to pedestrian safety events January –March 2015 by zip code

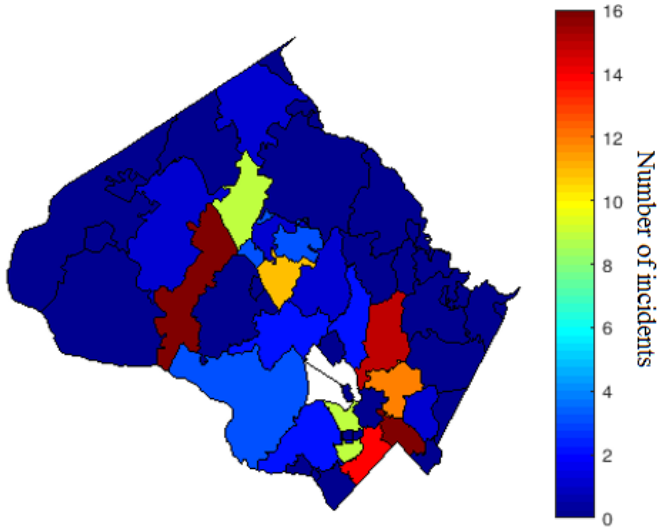


Fig. 7. Predicted traffic violations in Montgomery County, Maryland, related to pedestrian safety events January-March 2015 by zip code

Comparing output colors in the actual (top map) and predicted (bottom map) shows that our algorithm achieved excellent results. We split the 2015 data into four 3-month segments and show the map pair only for the first 3-month segment due to space limitations, even though mapped results throughout the year were equally good.

The high quality mapping results were confirmed by the .86 predictive accuracy shown in Table 1 of both the Probabilistic Graphical Model and the Poisson Regression. Based on the fact that the different data streams fed easily from the event model to the pedestrian safety algorithm shows that the event model is robust to at least three data types.

In creating an algorithm, we tried several statistical models for prediction. We evaluated results by creating a mapped output comparing actual pedestrian events to predicted events (Fig. 6 and 7), and by using the accuracy statistic averaged over a year of predictions (Table 1). The accuracy match between our prediction model and the real world data shows our service using data from our event model is accurate to .86 using either Probabilistic Graphical Models or a Poisson Regression. We can extend the algorithm to predict time as well as location.

Our pedestrian safety algorithm is limited by the fact that we do not have a record of pedestrian safety incidents in the county for 2015 that is entirely complete since we are limited to data that is available publically. However, it is strengthened by the fact that we found event instances to a high degree of accuracy. In a real-time application, there will likely be additional errors identifying event instances.

F. How government officials could use the system

Data from the Montgomery County, Maryland open government stream in JSON/xml first would be set to flow into the application. We would create a web-based interface for the application that would be active around the clock.

Users could examine the county map by zip code in the interface to see which areas in the county are having pedestrian incidents at that time (as in Fig. 7), and can then make informed decisions where county resources should be deployed to heighten safety. Our experiments have shown that the prediction algorithm is up to 86% accurate, and so the officials should feel confident that the resources would be deployed effectively. It would straightforward for us to add analytics to the interface as well, so that the officials could see trends over time.

V. Discussion

A. Semantic model for event processing architecture: toward a standard

Numerous event models in the literature vary with respect to required data fields and the role of ontologies. It is rare to find research that discusses deploying an event model: how to get data in and out, and how to handle missing data, for example. That makes it impossible to "plug in" other event models into our architecture for comparison. This paper has covered both the data modeling and some flexibles rules to guide deployment of our event model in a city service.

Our model requires a minimum of metadata fields, and it is efficient, minimizing the amount of network traffic by linking a continuous time dependent data stream to non-time dependent meta-information stored on an event server or cloud.

Overlap among different types of data sources should be handled by an upper level ontology such as DOLCE. We made the data field types in our model align with DOLCE categories in order to facilitate getting data from different streams into the same model.

A one-set-of-metadata-fits-all model would be more convenient than ours, but it would not necessarily capture what each particular service requires. That is why some thought will still be required to add parameters to suit the service being implemented.

Relying on a standard event model will make it easier for developers to build Complex Event Processing systems. In that our event model is generic in suiting a variety of data streams and a variety of domains, we propose it as basis for an event model standard.

B. Event model in Complex Event Processing architecture

Event models in complex event processing diagrams are ubiquitous. Actual systems that perform event processing are less common, and in fact, input of heterogeneous data fields into a single system is still coming into focus. The event model fits different data types into the same framework, thus making data integration easier.

Getting data into the model for highly structured can be as simple as matching metadata field names. But when the data flow does not include metadata, or when it is a natural language stream -- as in the police reports used here -- Natural Language Processing tools will employ data extraction. The most recent Named Entity Recognition (NER) tools to extract person, organization, location, and other objects that are proper nouns, are well advanced, achieving accuracies in the 90% range.

C. Publically-available, updated Internet of Things testbed

We applied Internet of Things data to help solve an actual problem: determine which areas of Montgomery County, Maryland require more monitoring and local resources to improve pedestrian safety. We used that algorithm to test our event model.

Few Internet of Things data sets at the time of writing are publically available for experimentation. It is not surprising, therefore, that we found that few of the research papers on event models use real-world data. Many papers are entirely theoretical, and some use synthetic data sets which were created or assembled from the web for the purposes of demonstrating a point.

Internet of Things data sets are obscured from a general web search because they are not referred to as Internet of Things, or sensor data – the keywords we use in computer science literature. The scientific community should look for human and sensor data relevant for Internet of Things research in open city projects. Our data come from Maryland, and New York has an even broader data set,[8] and the Open Data Portal has data from around the world.[9]

VI. Conclusion

An event model with an ontology to control word descriptions will bridge smart cities services by reusing data streams across applications. This approach will make applications more financially viable and expand the potential number and scope of smart city services. Our event model should be tested with more data stream types. Even better to test the event model would be to create services complementary to this in pedestrian safety to ensure that the same data can be used in multiple ways.

Acknowledgment

We are grateful for Maxence Lefort's advice on the design of our event model. The work was done in conjunction with the Information Technology Laboratory of the National Institute of Standards and Technology. The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

[1] Alevizos, E. Skariatidis, A., Artikis, A., Paliouras, G. (2015). Complex event recognition under uncertainty: a short survey.

[8] https://data.ny.gov/

[9] dataportals.org/search

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Workshop Proceedings of the EBDT/ICDT 2015 Joint Conference, March 27, 2015, Brussels, Belgium. [7 p.]

[2] Anicic, D., Fodor, P. Rudolph, S., Stojanovic, N. (2011). EP-SPARQL: a unified language for event processing and stream reasoning. In S. Srinivas, K., Ramamrithan, A. Kumar, M.P. Ravindra, E. Bertino, and R. Kumar Eds. *Proceedings of the 20th International Conference on World Wide Web WWW '11*, 635-644.

[3] Baekgaard, L, (2002) Event modeling in UML. Issues & Trends of Information Technology Management in Contemporary Organizations. Ed. Mehdi Khosrow-Pour (Information Resources Management Association, USA) [3 p.]

[4] Chen, H., Finin, T., & Joshi, A. (2005). The SOUPA ontology for pervasive computing. In Ontologies for agents: Theory and experiences (pp. 233-258). Birkhäuser Basel.

[5] Chiu, O. (2014). Announcing Azure Stream Analytics for real-time event processing. https://azure.microsoft.com/en-us/blog/announcing-azure-stream-analytics-for-real-time-event-processing/. Last time accessed: 26/02/2016.

[6] Cugola, G., Margara, A. (2010). TESLA: a formally defined event specification language. In *Proceedings of the Conference on Distributed Event-Based Systems (DEBS),* Cambridge, United Kingdom, 50-61.

[7] Doerr, M, Ore, C.E. & Stead, S. The CIDOC conceptual reference model: a new standard for knowledge sharing. In Conceptual modeling. Australian Computer Society, Inc., 2007.

[8] Ekin, A, Tekalp, A. M, & Mehrotra, R. Integrated semantic-syntactic video modeling for search and browsing. IEEE Transactions on Multimedia, 6(6), 2004.

[9] Fowler, C., & Qasemizadeh, B. (2009). Towards a common event model for an integrated sensor information system. In 1st International Workshop on the Semantic Sensor Web (SemSensWeb), [13 p.]

[10] Francois, A. R. J., Nevatia, R, Hobbs, J, & Bolles, R. C. VERL: An ontology framework for representing and annotating video events. IEEE MultiMedia, 12(4), 76-86.

[11] Fujitsu Develops Distributed and Parallel Complex Event Processing Technology that Rapidly Adjusts Big Data Load Fluctuations Online. Available: http://www.fujitsu.com/global/news/pr/archives/month/2011/20111216-02.html. Last time accessed: 26/02/2016.

[12] Gyrard, A., Serrano, M., Atemezing, A. (2015) Semantic web methodologies, best practices, and ontology engineering applied to Internet of Things. *IEEE Second World Forum on Internet of Things, Milan, Italy*, 412-217.

[13] Jiang, Y., Xu, Z., Wang, X. (2014). The construction of ontology in the area of traffic violations. J.J. Park et al (Eds). *Future Information Technology. Lecture Notes in Electrical Engineering 309*, 379-384.

[14] Kostis, K. and Katasonov, A. (2013). Semantic interoperability on the Internet of Things: The Semantic Smart Gateway Framework. *International Journal of Distributed Systems and Technologies 4*(3), 47-69.

[15] Kress, J., Maier, B., Normann, H., Schmeidel, D., Schmutz, G., Trops, B., Utschig-Utschig, C., Winterberg, T. (2016). Event-Driven SOA. http://www.oracle.com/technetwork/articles/soa/ind-soa-events-2080401.html. Last time accessed: 26/02/2016.

[16] Ma, M., Wang, P., Yang, J., Li, C., (2015). OntoEvent: An ontology-based event description language for semantic complex event processing. J. Li and Y. Sun (Eds). *WAIM 2015, LNCS 9098*, 448-451.

[17] Pongpaichet, S., Singh, V.K., Gao, M., Jain, R., (2013). EventShop: Recognizing situations in web data streams. *WWW2013 Companion, May 13-17, 2013, Rio de Janeiro, Brazil*, 1359-1367.

[18] Raimond, Y. Abdallah, S., Sandler, M., Giasson, F., (2007). The music ontology. Austrian Computer Society, [6 p.] http://raimond.me.uk/pubs/Raimond-ISMIR2007-Submitted.pdf

[19] Scherp, A., Franz, T. Saathoff, C. and Staab, S. (2009). F—A Model of Events based on the foundational ontology DOLCE+DnS Ultralite. *Knowledge Capture, '09, September 1-4, Redondo Beach, California*, 137-144.

[20] Selman, D., Ritchie, A., (2015). Event Driven Architecture and Decision Management. IBM's Operational Decision Manager. Available:https://developer.ibm.com/odm/docs/odm-capabilities/odm-advanced-decision-server-insights/event-driven-architecture-and-decision-management/. Last time accessed: 02/26/2016.

[21] Shaw, R., Troncy, R., Hardman, L. (2009). LODE: Linking Open Descriptions of Events in *'The Semantic Web*', Springer Berlin / Heidelberg, pp. 153-167.

[22] Shanahan, M. (1999). The event calculus explained. In Artificial intelligence today (pp. 409-430). Springer Berlin Heidelberg.

[23] Smirnov, A., Kashevnik, A., Shilov, N., Balandin, S., Oliver, I., Boldyrev, S. (2011). Development of the on-the-fly ontology matching model for smart spaces. In *Consumer Communications and Networking Conference (CCNC), 9-12 January, Las Vegas, Nevada, U.S.A.,* 808-809.

[24] Stühmer, R., Vergianadis, Y., Alshabani, I. Morsellino, T., Aversa, A. (2013). PLAY: Semantics-based event marketplace. *14th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE Sept 2013, Dresden, Germany*.

[25] Uma, V., and Aghila, G. (2014). Event order generation using Reference Event based qualitative Temporal (REseT) relations in Time Event Ontology. *Central European Journal of Computer Science 4*(1): 12-29.

[26] Wang, X, Mamadgi, S, Thekdi, A., Kelliher, A., & Sundaram, H. Eventory - an event based media repository. In Semantic Computing. IEEE, 2007.

[27] Westermann, U, and Jain, R. (2007). Toward a common event model for multimedia applications. IEEE multimedia 14(1): 19-29.

[28] Zhong, Z., Liu, Z., Li, C., & Guan, Y. (2012). Event ontology reasoning based on event class influence factors. International Journal of Machine Learning and Cybernetics, 3(2), 133-139.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Appendix: Comparison chart for event models *

	Processing				Required fields								Other requirements	
Event Model	Raw data into model	Handle missing data	Handle uncertainty	Data types	Static/dynamic	Action/verb/event	Participant/actor/subject/Agent	Object	Time	Location	Device associated	Device affected	Ontology	Processing language
Event Modeling in UML (Baekgaard, 2002) [3]	ns	ns	ns	ns	ns	Y	Y	Y	ns	ns	ns	ns	no	no
Eventory (Wang et al, 2007) [26]	Y	ns	ns	ns	ns	Y	Y	ns	Y	Y	ns	ns	no	no
Event ML (IPTC, release 2014)	ns	ns	ns	ns	ns	Y	Y	ns	Y	Y	ns	ns	no	XML
SsVM (Ekin et al, 2004) [8]	ns	ns	ns	video	ns	Y	Y	Y	Y	Y	ns	ns	no	SQL
Event E (Westermann et al, 2007) [27]	ns	ns	Y	Multi-media	ns	Y	Y	ns	Y	Y	ns	ns	no	no
Event F (Scherp et al, 2009) [19]	ns	Y	ns	Any	ns	Y	Y	ns	Y	Y	ns	ns	DOLCE+DnS	RDF/XML
OntoEvent (Ma et al, 2015)	ns	ns	ns	Location, Common data types	ns	Y	Y	ns	Y		ns	ns	Created by them	OntoEvent lang.
REseT (Uma et al, 2014) [25]	Y	ns	ns	Common data types	ns	Y	Y	ns	Y	Y	ns	ns	Created by them	DL
Common Event Model (Fowler et al, 2009) [9]	ns	ns	ns	Location, common data types	ns	Y	Y	Y	Y	Y	Y	ns	Created by them	RDF/XML
Event ontology (Zhong et al, 2012) [28]	ns	ns	ns	Common data types	ns	Y	Y	ns	Y	Y	ns	ns	Created by them	RDF/XML
VERL (Francois et al, 2005) [10]	ns	ns	ns	Multi-media	ns	Y	Y	Y	Y	Y	ns	ns	VEML 2.0/OWL	VEML/OWL-DL
CIDOC CRM (Doerr et al, 2007) [7]	ns	ns	ns	ns	ns	ns	Y	ns	Y	Y	ns	ns	ISO 21127	XML
SOUPA (Chen et al, 2005) [4]	ns	ns	ns	Common data types	ns	Y	Y	ns	Y	Y	Y	ns	COBRA-ONT/OWL	RDF/XML
LODE (Shaw et al, 2009) [21]	Y	ns	Y	ns	ns	ns	Y	ns	Y	Y	ns	ns	Created by them	RDF/XML
EventOntology (Raimond et al, 2007) [18]	Y	ns	ns	Multi-media	ns	Y	Y	Y	Y	Y	ns	ns	Created by them	RDF/XML
Event Calculus (Shanahan, 2001) [22]	ns	ns	ns	ns	ns	Y	ns	ns	Y	Y	ns	ns	no	Some logic language
Ours (Kotevska et al, 2016)	Y	Set by coder	Y	any	Y	Y	no	no	Y	Y	Y	no	any	any

*ns = not specified, Y = yes

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The Bugs Framework (BF): A Structured Approach to Express Bugs

Irena Bojanova
NIST
Gaithersburg, USA
irena.bojanova@nist.gov

Paul E. Black
NIST
Gaithersburg, USA
paul.black@nist.gov

Yaacov Yesha
NIST; *UMBC*
Gaithersburg, USA; Baltimore, USA
yaacov.yesha@nist.gov

Yan Wu
BGSU
Bowling Green, USA
yanwu@bgsu.edu

***Abstract*—To achieve higher levels of assurance for digital systems, we need to answer questions such as does this software have bugs of these critical classes? Do two software assurance tools find the same set of bugs or different, complimentary sets? Can we guarantee that a new technique discovers all problems of this type? To answer such questions, we need a vastly improved way to describe classes of vulnerabilities and chains of failures. We present the Bugs Framework (BF), which raises the current realm of best efforts and useful heuristics. Our BF includes rigorous definitions and (static) attributes of bug classes, along with their related dynamic properties, such as proximate, secondary and tertiary causes, consequences and sites. The paper discusses the buffer overflow class, the injection class and the control of interaction frequency class, and provides examples of applying our BF taxonomy to describe particular vulnerabilities.**

Keywords—software weaknesses; bug taxonomy; attacks.

I. Introduction

The medical profession has an extensive, elaborate vocabulary to precisely name muscles, bones, organs and diseases. When a doctor says that a comatose patient has a left temporal lobe epidural hematoma, the intention is to enlighten, not obfuscate. In the software profession, many efforts have developed terms to discuss software, faults, failures and attacks, such as the Common Weakness Enumeration (CWE) [1] and Landwehr et. al. Taxonomy of Computer Program Security Flaws [2], but much work remains.

We want to more accurately and precisely define software bugs or vulnerabilities. Consider that adding "canary" values around arrays detects some buffer overflows while using address layout randomization mitigates others. A precise, orthogonal nomenclature can state exactly which classes of buffer overflows each approach handles. We can also clearly state the classes of bugs that a tool can find and more easily determine if two tools generally find the same set of bugs or if they find different, complimentary sets.

The ancient Greeks used the terms element and atom, and Aristotle proposed that all matter is a mixture of earth, air, fire or water. In the Middle Ages, alchemists made lists of materials, such as alcohol, sulfur, mercury and salt. Through centuries of experimentation and development of scientific principles, we now have Mendeleev's Periodic Table of Elements, see Fig. 1. Just as the structure of the periodic table reflects the underlying atomic structure, we are developing a taxonomy dictated by the "natural" organization of software bugs, while using as stepping stones known bugs enumerations, compendia and collections.

Over the course of history, science has developed many different organizational structures. Linnaeus' taxonomy categorizes living things into a hierarchy of Domain, Kingdom, Phylum, Class, Order, Family, Genus and Species. It allows comprehension of the diversity of life forms and codifies understanding that some animals are close in their evolutionary history. The Geographic Coordinate System specifies any location on Earth using latitude, longitude and elevation. The Dewey Decimal Classification system allows new books and whole new subjects to be placed in reasonable locations in a library for easy retrieval based on subject. Fingerprints are

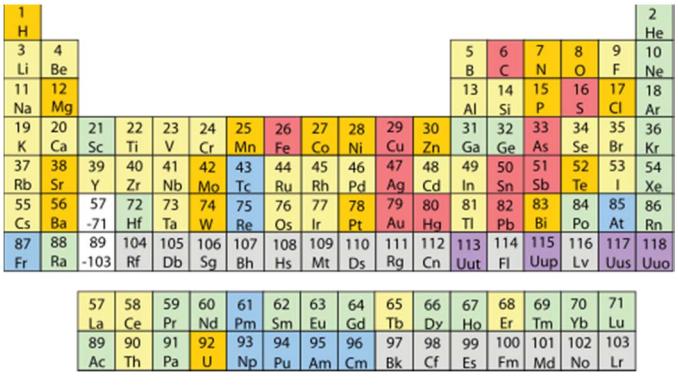


Fig. 1. Periodic Table of Elements: antiquity, Levoisier 1789, Mendeleev 1869, Deming 1923, Seaborg 1945, up to 2000, to 2012[1].

Disclaimer: Certain trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology (NIST), nor does it imply that they are necessarily the best available for the purpose.

[1] By Sandbh - Wikimedia Commons., CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=31017351

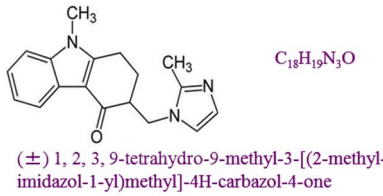


Fig. 2. Three ways to describe Zofran ODT.

classified using loops, whorls and arches and retrieved based on minutia. Chemists have a detailed system beyond the periodic table to describe chemicals. For instance, they have several different systems of rendering molecules, which are three dimensional, to emphasize aspects that are more important in different contexts, see Fig. 2.

Finally, all integers[2] have unique prime factors. Analogously, we seek to factor software weaknesses into their constituent components, thereby gaining the understanding to organize these components in their most naturally-occurring categories and structure. We aim for the most accurate, precise and intuitive way to describe software bugs.

To paraphrase William Thomson, Baron Kelvin, "when you can measure what you're speaking about, and express it [in precise terms], you know something about it; but when you cannot, your knowledge is of a meager and unsatisfactory kind: it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced it to the stage of *science*." [3]

In this paper, we first discuss existing software weaknesses enumerations, patterns and templates. Then we present our Bugs Framework (BF) with its four main areas: causes, attributes, consequences and sites. To make sure that BF applies to all classes of bugs, we began with three quite disparate classes: buffer overflows, injections and control of frequency of interactions. Buffer overflow occurs primarily in C and is low-level; injection relates strongly to the language in which the command string is interpreted, and control of frequency interactions requires reference to a user-level policy to set limits. For each class, we provide a definition and the BF taxonomy, which includes the sites in code where they may be found. We also provide examples of applying the taxonomy to describe particular vulnerabilities and list corresponding classes from other weaknesses collections. The final section summarizes our work and discusses the benefits from our BF as well as our future plans. Our goal is for the BF to become the software developers' and testers' "Best Friend."

II. Existing Enumerations, Patterns and Templates

The Common Weakness Enumeration (CWE) [1] is an "encyclopedia" of over 600 types of software weaknesses. Some of the classes are buffer overflow, directory traversal, OS injection, race condition, cross-site scripting, hard-coded password and insecure random numbers. CWE is a widely-used compilation, which has gone through many iterations. Many tools and projects are based on it. Each CWE has a variety of information, such as description summary, extended description, white box definition, consequences, examples, background details and other notes, recorded occurrences (Common Vulnerabilities and Exposures or CVE [4]), mitigations, relations to other CWEs, and references.

CWEs are a rich source of material for software developers and superior to anything that existed before. However, for very formal, exacting work, CWE definitions are often inaccurate, imprecise or ambiguous, and the various definitions within one CWE can be inconsistent. Each CWE bundles many stages, such as likely attacks, resources affected and consequences. The coverage is uneven, with some combinations of attributes well represented and others not appearing at all. An extreme instance is path traversal. There are a dozen CWEs for path traversal, each one having a specific combination of relative or absolute paths, forward or backward slashes – singly or repeated, between one and three directory steps, and two or more dots, which indicate the parent directory.

Another example is buffer overflows. CWE-121 [5] is write outside of a buffer on the stack, CWE-122 is write outside of a buffer in the heap, CWE-127 is read before the beginning of a buffer and CWE 126 is read after the end of a buffer. But there are no CWEs specifically for read outside a buffer on the stack vs. in the heap. The description summary of CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer is "The software performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer." Note that "read from or write to a memory location" is not explicitly tied to the buffer! Most humans would, of course, assume that it means the software can access through a buffer a memory location that is not allocated to that buffer.

Software Fault Patterns (SFP) [6] are a clustering of CWEs into related weakness categories. Each cluster is factored into formally defined attributes, with sites ("footholds"), conditions, properties, sources, sinks, etc. This work overcomes the problem of combinations of attributes in CWE. For instance, Table 1 shows how SFP factored attributes are more clear than the irregular coverage of CWEs.

SFP is an excellent advance, but does not tie fault clusters to causes or chains of fault patterns nor to consequences of a particular vulnerability. In addition, since they were derived from CWEs, more work is needed for embedded or mobile concerns, such as, battery drain, physical sensors (e.g. Global Positioning System (GPS) location, gyroscope, microphone, camera) and wireless communications.

Another source of organization of weaknesses is Semantic Templates (ST). "A semantic template is a human and machine understandable representation of the following: 1) software faults that lead to a weakness; 2) resources that a weakness affects; 3) weakness attributes; and 4) consequences/ failures resulting from the weakness." [7] Semantic Templates factor out chains of causes, resources and consequences that are present in CWEs. For instance, Fig. 3 shows phrases in the description summary, extended description and common consequences of CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow'), labeled according to the phases called out by Semantic Templates.

Details on the relevant body of knowledge that consolidates CWE, including the SFP and the ST efforts is presented in [8].

[2] Greater than one.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

TABLE I. SFP FACTORED ATTRIBUTES OF BUFFER OVERFLOW CWES

CWE \ Attribute	Location		Access kind		Boundary exceeded	
	heap	*stack*	*read*	*write*	*lower*	*upper*
119: Improper Restriction of Operations within Bounds of Buffer	√	√	√	√	√	√
120: Buffer Copy without Checking Size of Input	√	√		√	√	√
121: Stack Overflow		√		√	√	√
122: Heap Overflow	√			√	√	√
123: Write-what-where condition	√	√		√	√	√
124: Buffer Underwrite	√	√	√	√	√	
125: Out-of-bounds read	√	√	√		√	√
126: Buffer Overread	√	√	√			√
127: Buffer Underread	√	√	√		√	

Landwehr et. al. created a taxonomy of security flaws in programs [2]. The taxonomy has three aspects: genesis, that is, how it originated, time of introduction and location. Each aspect is further divided into subcategories. The main focus of the taxonomy seems to be how flaws originated and is aimed at a higher, system level. It does not include details enabling automated detection in code, proving the efficacy of mitigation techniques or deriving possible consequences.

III. THE BUGS FRAMEWORK (BF)

Just as integers can be factored into prime numbers or molecules can be decomposed into constituent atoms, we break down information in CWEs, SFPs, and other compendia and collections into basic, orthogonal components.

We organize them into meaningful structures and identify rules of composition. We use this compilation in several ways in order to validate it and demonstrate its utility. We elucidate known vulnerabilities, accurately and precisely defining the classes of bugs reported by assurance tools and document in exactly what situation various software assurance techniques are efficacious. We believe this compilation may also guide development of techniques to cover gaps.

The BF comprises four main areas: causes, attributes, consequences and sites of bugs. The causes and consequences are well represented with a directed graph. *Causes* include implementation mistakes, conditions, preceding weaknesses and circumstances that bring about the fault. Some of the causes are nested hierarchically. The identifying or distinguishing *attributes* are the next general area.

Some assurance techniques or mitigation approaches may work for a fault with certain attributes, but not for the same general kind of fault that has other attributes. Each attribute is an enumeration of possible values. Lists of attributes also open the opportunity to more formally define and reason about them.

Note that the attributes describe an event, not the site in code that gives rise to the event.

We want to be able to forecast possible *consequences* of different kinds of faults. Knowing what consequences might occur allows risk estimation and determination of best mitigation strategies.

CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')

Description Summary: The program copies an input buffer to an output buffer without verifying that the size of the input buffer is less than the size of the output buffer, leading to a buffer overflow.

Extended Description: A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold, or when a program attempts to put data in a memory area outside of the boundaries of a buffer. The simplest type of error, and the most common cause of buffer overflows, is the "classic" case in which the program copies the buffer without restricting how much is copied.

Common Consequences: Buffer overflows often can be used to execute arbitrary code, which is usually outside the scope of a program's implicit security policy. This can often be used to subvert any other security service. Buffer overflows generally lead to crashes. Other attacks leading to lack of availability are possible, including putting the program into an infinite loop.

Fig. 3. Phrases in CWE-120 descriptions labeled according to ST phases. Blue is software faults. Yellow is a weakness. Green is resource or location. Red is consequences.

Finally, we describe the *sites* or locations in code where the bug might occur under circumstances indicated by the causes.

A site is a location in code where a weakness might be For instance, every buffer access in a C program is a site where buffer overflow might occur if the code is buggy. In other words, sites for a weakness are places that must be checked for that weakness. [9] The determination of sites depends only on local information. That is, global or flow-sensitive information is not needed to determine where sites are in code.

For example, the following code comes from Software Assurance Reference Dataset (SARD) [10] case 62 804. It has one site of writing to an array, `data[i] = ...`, which needs to be checked for a write-outside-array bug. There is also one site of reading from an array, `source[i]`, where the program might read outside the array if there is a bug.

```
for (i = 0; i < 10; i++) {
    data[i] = source[i];
}
```

In addition, the code has sites of possible uninitialized variable, every place that `i` is used, and a possible integer overflow site, `i++`. Notice that the assignment statement in the body of the loop has several sites.

This statement-level definition of site not always applies. When a C programmer uses the `strcpy` library function, it does not get enough information to check for a buffer overflow. Similarly the Structured Query Language (SQL) processor cannot determine that the programmer never intended queries like "name = Henry or 1=1" to be always true. The site is the last or lowest level of code execution outside library functions or utilities. This is the final chance the programmer had to avoid the fault. In other words, sites of a bug are places in the code that should be checked for that class of bug.

Following are one section for each of three classes of bugs from our BF: buffer overflows, injections and control of frequency of interactions. In each section, we give a definition

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

of the class and our taxonomy, including related sites. Following that we provide examples and related classes from other collections, such as CWEs and SFPs.

IV. Buffer Overflow Class – **BOF**

A. *Definition*

We define Buffer Overflow (BOF) as:

> *The software can access through an array a memory location that is outside the boundaries of that array.*

Often referred to as a "buffer," an array is a contiguously allocated set of objects [11], called elements. An array has a definite size—a definite number of objects are allocated to it. The elements are of same data type and are accessed by integer subscripts.

If the software can utilize the array name (more generally, array handle or pointer to array elements) to access any memory other than the allocated objects, it falls into this class.

B. *Taxonomy*

Fig. 4 depicts BOF causes, attributes and consequences. The ACI cluster of consequences appears also in Fig. 5 and Fig. 6.

The graph of causes for BOF shows that there are only two proximate causes of buffer overflows: amount of data exceeds the array size or there is a wrong index or pointer. Those two causes have preceding causes that may lead to them. Note that "Data Exceeds Array" could be the result of an "Array Too Small" or result of "Too Much Data." The former sub-cause means that an incorrectly small array has been allocated and because of that the destination is too small. The latter sub-cause means that incorrectly large amount of data has been accessed and because of that the data is too big.

The attributes of BOF are:

Access – Read, Write.

Boundary – Below, Above. This indicates which end of the array is violated. Synonyms for boundary are side or bound. The terms before, under or lower may be used instead of below. The terms after, over or upper may be used instead of above. Outside indicates that the boundary is unknown or it doesn't matter.

Location – Heap, Stack. This indicates what part of memory the array is allocated in. It may matter since violations in the stack may affect program execution flow, while violations in the heap typically only affect data values. Other architectures may have other locations that are significant. For instance, Intel architecture also has Bss, Data and Code (text).

Magnitude – Small, Moderate, Far. This is how far outside the boundary the violation extends. Small means just barely outside, e.g. one to a few bytes beyond the end, moderate is something like eight bytes to dozens, and far is hundreds, thousands or more.

These distinctions in the magnitude attribute are important because some violation detection techniques or mitigation techniques, such as canaries or allocating a little extra space, are only useful if the magnitude is small.

Data Size – Little, Some, Huge. This is how much data is read or written beyond the boundary. Like in magnitude, these distinctions are important in some cases. For instance, Heartbleed [12] would not have been a severe problem if it just exfiltrated a little data. The fact that it may exfiltrate a huge amount of data greatly increases the chance that very important information will be leaked.

Reach – Continuous, Discrete. This indicates whether the access violation was preceded by consecutive access of elements starting within the array (continuous) or just an access outside of the array (discrete). Typically string accesses or array copies handle a continuous set of array elements, while a vagrant array index only reads or writes one element.

Note that any of the attributes may be "any," "don't care" or "unknown." For instance, strict bounds checking is equally effective regardless of the location, magnitude, data size or reach of the violation. Keeping return addresses in a separate stack

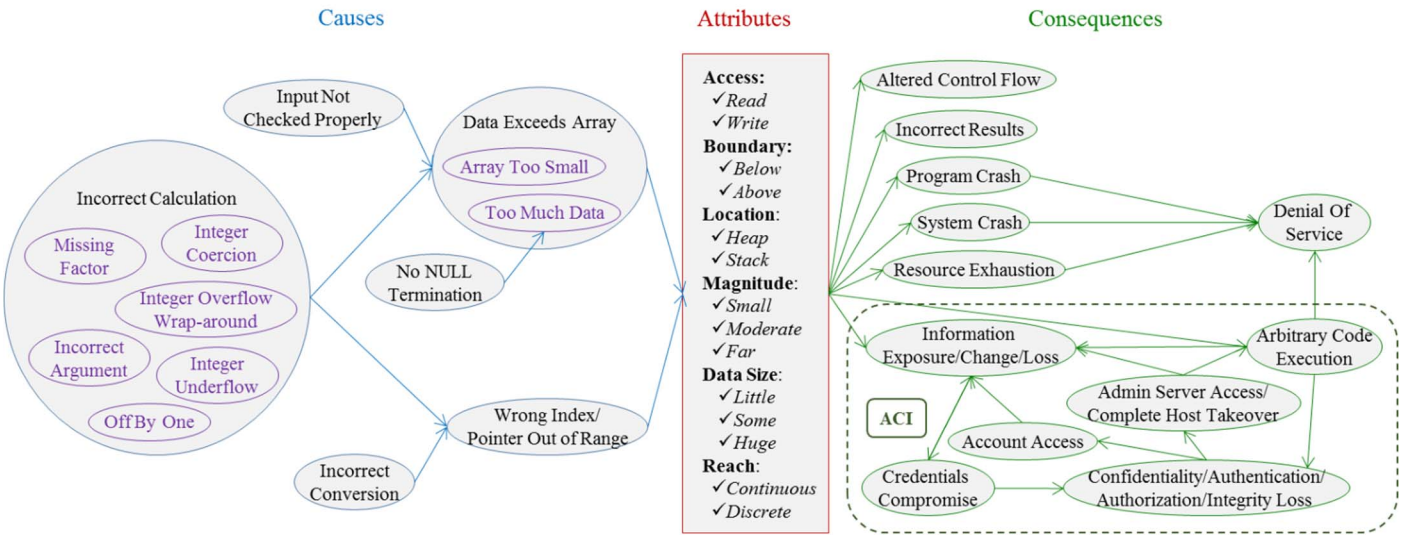


Fig. 4. The Buffer Overflow (BOF) class represented as causes, attributes and consequences. (The ACI cluster is the same in all classes.)

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

helps prevent problems occurring from write accesses when the array location is the stack.

The values for the access, boundary, location, magnitude, and reach attributes were listed earlier by Kratkiewicz [13], although they were discovered independently through our analysis of the buffer overflow CWEs. Some additional attributes from [13] that might be relevant to our BOF taxonomy are: Data Type, for instance int, float or Boolean, and Container, for instance, the array is in a struct or record.

Note that in the graph of consequences in Fig. 4, "Resource Exhaustion" refers to Memory and CPU.

In the C language, sites where a buffer overflow may occur are the use of `[ ]` or unary `*` operators with arrays. Sites also include the use of string library functions as `strcpy` or `strcat`.

C. Examples

1) CVE-2014-0160 – Heartbleed

This vulnerability is listed in [12] and discussed in [14]. Our BF description is:

Input not checked properly leads to too much data, where a huge number of bytes are read from the heap in a continuous reach after the array end, which may be exploited for exposure of information that had not been cleared.

2) CVE-2015-0235 – Ghost

This vulnerability is listed in [4] and discussed in [15]. Our BF description is:

Incorrect calculation, (specifically missing factor) leads to array too small, where a moderate number of bytes are written to the heap in a continuous reach after the array end, which may be exploited for arbitrary code execution, leading to denial of service.

3) CVE-2010-1773 – Chrome WebCore

This vulnerability is listed in [4] and discussed in [16, 17, 18, 19, 20, 21]. Our BF description is:

Incorrect calculation, (specifically off by one) leads to a wrong index, where a small number of bytes are read from the heap in a discrete reach before the array start, which may be exploited for information exposure, arbitrary code execution or program crash, leading to denial of service.

D. Related CWEs, SFP and ST

CWEs related to BOF are: CWE-119, 120, 121, 122, 123, 124, 125, 126, 127, 786, 787 and 788. The only related SFP cluster is SFP8 Faulty Buffer Access under Primary Cluster: Memory Access [22]. The corresponding ST is the Buffer Overflow Semantic Template [23].

V. Injection Class – **INJ**

A. Definition

We define Injection (INJ) as:

Due to input with language-specific special elements, the software can assemble a command string that is parsed into an invalid construct.

In other words, the command string is interpreted to have unintended commands, elements or other structures.

B. Taxonomy

Fig. 5 depicts INJ causes, attributes and consequences.

The attributes of INJ are:

Language – Database Query, Regular Expression, Command, Markup, Script. This indicates the language in which the command string is interpreted. Database query language could be SQL. Command language could be Bash. Markup language could be XML/Xpath. HTML Scripting language could be PHP, CGI.

Special Element – Query Elements, Header Separators, Scripting Elements, Format Parameters, Path Traversals, Wildcards, Metacharacters. These could be assembled with other elements to form malicious structures such as queries, scripts and commands. Query elements include strings

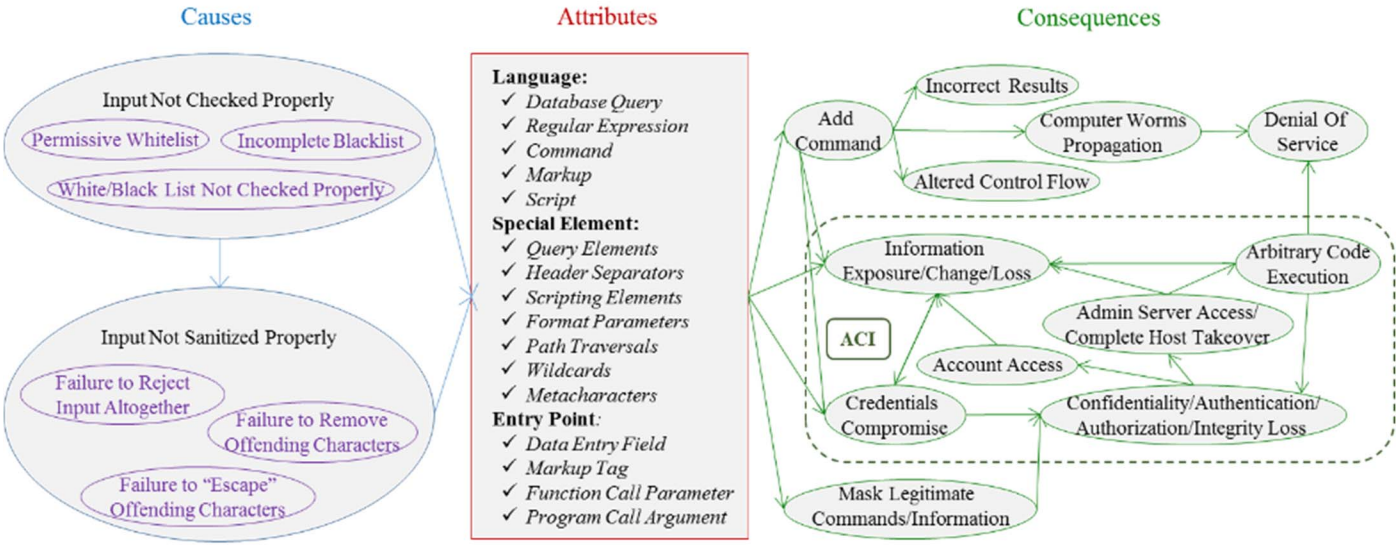


Fig. 5. The Injection (INJ) class represented as causes, attributes and consequences. (The ACI cluster is the same in all classes.)

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

delimiters ‘ or “ or words such as ‘and’ or ‘or’. Header separators include carriage return/line feed. Scripting elements are such as < or > or &. Format parameters are such as %c or %n. Path traversals elements include .. or \. Metacharacters are back tick (`) or $ or &.

Entry Point – Data Entry Field, Scripting Tag, Markup Tag, Function Call Parameter, Procedure Call Argument. This indicates where the input came from.

Note that in the graph of consequences on Fig. 5, “Arbitrary Code Execution” concerns any instructions to the computer – compiled, interpreted by software, executed directly by hardware or combination.

Injection sites are typically not primitive operations in most languages. Sites are the library or utility functions that accept a command string for actions. In shell commands, command substitution is invoked with paired back quotes (`…`) or $(…). Command substitution executes a subshell, which opens the possibility of the string to be interpreted with all the richness of the command line interpreter.

C. Examples

1) CVE-2007-3572 – Yoggie Pico

This vulnerability is listed in [4] and discussed in [25]; special elements are discussed in [26]. Our BF description is:

Input not checked properly (specifically incomplete blacklist) allows shell command injection through the “param“ function parameter in a CGI script using Shell metacharacters (specifically back ticks `), which may be exploited to add command, leading to arbitrary code execution.

Note that adding a command through Ping to change the root password enables eventual complete host takeover.

2) CVE-2008-5817

This vulnerability is listed in [4] and discussed in [27, 28]. Our BF description is:

Input not checked properly or input not sanitized properly allows SQL injection through the “username“ & “password“ fields in a PHP script using query elements (specifically single quote ‘ , the word or, and equality sign =), which may be exploited to mask legitimate SQL commands, leading to authentication compromise, admin server access and arbitrary SQL code execution.

3) CVE-2008-5734

This vulnerability is listed in [4] and discussed in [29, 30, 31]. Our BF description is:

Input not sanitized properly allows XSS web script or HTML injection through the IMG element of a generated HTML email, which may be exploited to add commands or for cookie-based authentication credentials compromise, leading to arbitrary code execution.

D. Related CWEs, SFPs and ST

CWEs related to INJ are CWE-74, 75, 77, 78, 80, 85, 87, 88, 89, 90, 91, 93, 94, 243, 564, 619, 643 and 652. Related SFPs are SFP24 and SFP27 under Primary Cluster: Tainted Input, and SFP17 under Primary Cluster: Path Resolution [22]. The corresponding ST is the Injection Semantic Template [32].

VI. Control of Interaction Frequency Class – **CIF**

A. Definition

We define Control of Interaction Frequency (CIF) as:

The software does not properly limit the number of repeating interactions per specified unit.

In physics, frequency is the number of occurrences of a repeating event per unit time [24]. Interactions in software could be also per event or per user.

B. Taxonomy

Fig. 6 depicts CIF causes, attributes and consequences.

The attributes of CIF are:

Interaction – Authentication Attempt, Book, Checkout, Register, Initiate. This indicates the type of interactions to be

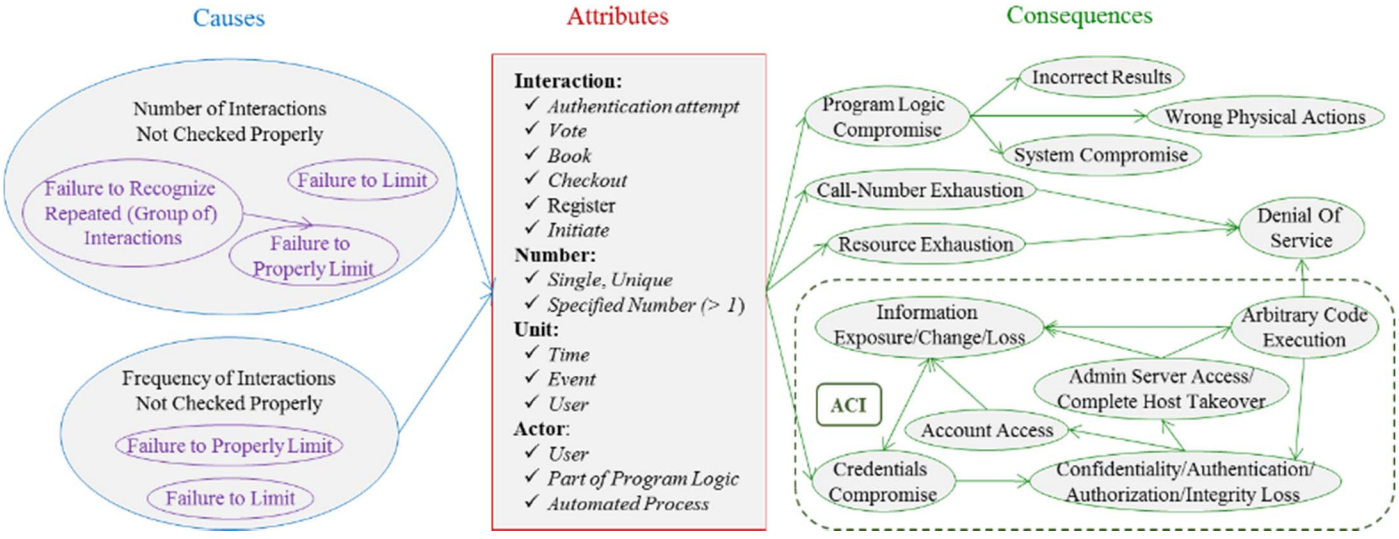


Fig. 6. The class Control of Interaction Frequency (CIF) represented as causes, attributes and consequences. (The ACI cluster is the same in all classes.)

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

controlled. Voting could be related to election, census, survey, referendum and ballot. Booking could be of tickets, hotel rooms or rental cars. Checkout could be of library books, hotel rooms or rental cars. Register could be for computer games. Initiate could be for message exchange.**Number –** Single, Unique; Specified Number (> 1). This indicates the maximum number of occurrences allowed.

Unit – Time Interval, Event, User. This indicates the specific unit per which the number of occurrences is controlled. Time Interval could be in seconds, in days, etc. Event could be election, authentication, on-line transaction to move funds, etc.

Authentication event is the sequence of authentication attempts arriving at a particular server, possibly with the same partial credential, from any source, that terminates by successful authentication or by blocking.

Actor – User, Part of Program Logic, Automated Process. This indicates who/what is performing the repeating interactions. User could be authenticated user, attacker. Part of program logic could be message exchange. Automated process could be virus, bot.

Note that in the graph of consequences in Fig. 6, "Credentials" concerns username or password, smart card and personal identification number (PIN), retina, iris, fingerprint, etc. "Resource Exhaustion" concerns memory, CPU or granted licenses.

Our taxonomy makes it abundantly clear that CIF is a "metaclass" in some senses. External policies must define for each system or application what constitutes an interaction, how many interactions should be allowed, and the unit. Each policy, then, defines a different class of CIF concerns.

Since the concept of interaction is so broad and high level, compared to most programming languages, no general description of what is a site is feasible. Each system or application must define its own concept of interaction. An interaction must then be mapped to some code that controls or authorizes said interactions. More importantly, since a failure may be the total lack of code to recognize and control frequency of interaction, there is often no particular line or even block of code that can be pointed out as missing the control code. An entire path may be indicated from the beginning of an interaction event, that is, an outside agent indicates desire to start an interaction, to the final chance in execution flow that code may refuse to authorize the event.

C. Examples

1) CVE-2002-0628

This vulnerability is listed in [4]. Our BF description is:

Failure to limit to a specified number the authentication attempts per authentication event by same or different user(s) may be exploited for credentials compromise (username or password) via brute force.

2) CVE-2002-1876

This vulnerability is listed in [4]. Our BF description is:

Failure to recognize repeated interactions that are rapid initiations of message exchange requests from authenticated users, leads to failure to properly limit them to a specified number per specified time interval, which may be exploited for resource exhaustion (consumption of all granted licenses) leading to denial of service.

3) CVE-2002-1018

This vulnerability is listed in [4]. Our BF description is:

Failure to limit the checkouts of a book to a single one per user may be exploited for resource exhaustion, leading to denial of service.

D. Related CWEs and SFP

CWEs related to CIF are CWE-799, 307 and 837. The related SFP cluster is SFP34 Unrestricted Authentication under the Primary Cluster: Authentication [22].

VII. Conclusions

A. Summary

We have shown a superior, unified approach. The presented Bugs Framework (BF) allows accurate, precise and unambiguous expression of software bugs or vulnerabilities. It can also be used to clearly explain the applicability and utility of different software quality or assurance techniques or approaches, which is demonstrated in the discussion of the magnitude and data size attributes of BOF.

This approach is a factoring and restructuring of information contained in CWEs, SFPs and STs, and thus benefits from the community's experience with their use. Instead of trying to match weakness classes that tools find to CWEs, usually far over- or under-generalizing, the BF can describe tool classes much more accurately, precisely and succinctly. Table 1 shows how this refinement approach allows clearer and more succinct descriptions. The BF consists of (1) causes arranged in a directed graph, (2) attributes of a software fault, (3) possible consequences of the fault, also in a directed graph and (4) possible sites in code, that is, locations that must be reviewed for possible faults. Causes and consequences may be hierarchical, too. For instance, "Data Exceeds Array" in BOF is either "Array Too Small" or "Too Much Data." "Input Not Checked Properly" in the INJ class is either "Permissive Whitelist" or "Incomplete Blacklist."

B. Benefits

With our BF practitioners and researchers can more accurately, precisely and clearly describe problems in software, discuss the classes of bugs that tools report or explain what vulnerabilities the proposed techniques prevent. Instead of adding more and more CWEs for every slight variant, types of weaknesses can be categorized unambiguously, allowing similarities and differences to be easily explored and examined. We believe that as CWEs migrate to using this kind of taxonomy, they will be easier to comprehend and avoid.

Those concerned with software quality, the reliability of programs and digital systems, or cybersecurity will be able to make more rapid progress now that they can more clearly label

the results of errors in software. Those responsible for designing, operating and maintaining computer complexes can communicate with more exactness about threats, attacks, patches and exposures.

C. Future Work

Although we demonstrate this approach on three disparate classes of weaknesses, much work remains. More examples, such as CVEs and tool classes, need to be expressed using this scheme to bring out facets that were overlooked or find better ways of organizing the information. Consequences need to be examined across all classes to better understand how, say, adding commands can lead to whole host takeover, regardless of the weakness allowing the addition. Chains of causes should be researched similarly. A concerted investigation of chains through particular attributes, drawing on existing work, should help clarify the relations between them. We also need to refactor many other bugs classes, which will turn up commonalities. We are building a web site at https://samate.nist.gov/BF/ which will have the latest information, such as guide books for classes.

As our BF covers more classes, existing taxonomies, like CWE, can start explaining their current entries with concise forms of our descriptions. Bug trackers can be enhanced to allow BF descriptions to be given. Many tool makers, such as static analyzers and bug trackers, already use CWEs. In the future, this use of CWEs can evolve to integrate BF into bug descriptions. For instance, a software assurance tool maker can find a CWE similar to the class of bugs that their tool can find and start with its BF description. The tool maker then can refine the BF description, changing enumerations of attributes until it matches their tool's class.

REFERENCES

[1] The MITRE Corporation. Common Weakness Enumeration (CWE). http://cwe.mitre.org.

[2] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi. A taxonomy of computer program security flaws, and examples. ACM Computing Surveys. vol. 26. no. 3. pp. 211–254. September 1994.

[3] W. Thomson, Baron Kelvin. Electrical units of measurement. Popular Lectures and Addresses. MacMillan. 1889. vol. 1. p. 73. A Lecture delivered at the Institution of Civil Engineers. May 3. 1883.

[4] The MITRE Corporation. Common Vulnerabilities and Exposures or (CVE). https://cve.mitre.org.

[5] The MITRE Corporation. Common Weakness Enumeration. CWE 121. https://cwe.mitre.org/data/definitions/ 121.html.

[6] N. Mansourov and D. Campara. System Assurance: Beyond Detecting Vulnerabilities. Morgan Kaufmann. 2010. pp. 176–188.

[7] Y. Wu, R. A. Gandhi, and H. Siy. Using Semantic Templates to Study Vulnerabilities Recorded in Large Software Repositories. Proc. 2010 ICSE Workshop on Software Engineering for Secure Systems. ser. SESS '10. New York, NY: ACM. 2010. pp. 22–28. http://doi.acm.org/10.1145/1809100.1809104.

[8] Y. Wu., I. Bojanova, and Y. Yaacov. They Know Your Weaknesses – Do You?: Reintroducing Common Weakness Enumeration. *CrossTalk* (The journal of Defense Software Engineering). Sept-Oct 2015. http://static1.1.sqspcdn.com/static/f/702523/26523304/1441780 301827/201509-Wu.pdf.

[9] P. E. Black and A. Ribeiro. SATE V Ockham Sound Analysis Criteria. National Institute of Standards and Technology (NIST). NIST IR 8113. March 2016. http://dx.doi.org/10.6028/NIST.IR.8113.

[10] Software Assurance Reference Dataset (SARD). https://samate.nist.gov/SARD.

[11] ISO/IEC 9899:2011 programming languages - C, Committee Draft—April 12, 2011 N1570. ISO/IEC Joint Technical Committee JTC 1, Information technology, Subcommittee SC 22, Programming languages, their environments and system software interfaces. Working Group WG 14 – C. Tech. Rep. 2011.

[12] The MITRE Corporation. CVE-2014-0160. https://cve.mitre. org/cgi-bin/cvename.cgi?name=CVE-2014-0160.

[13] K. Kratkiewicz. Evaluating Static Analysis Tools for Detecting Buffer Overflows in C Code. Master's thesis. Harvard University, Cambridge, MA. March 2005. https://www.ll.mit.edu/mission/cybersec/publications/publication-files/full_papers/KratkiewiczThesis.pdf.

[14] S. Cassidy. Diagnosis of the OpenSSL Heartbleed Bug. https://www.seancassidy.me/diagnosis-of-the-openssl-heartbleed-bug.html.

[15] Openwall. Qualys Security Advisory CVE-2015-0235 - GHOST: glibc gethostbyname buffer overflow. http://www.openwall. com/lists/oss-security/2015/01/27/9.

[16] R. Gandhi. Buffer Overflow Semantic Template: CVE-2010-1773. http://faculty.ist.unomaha.edu/rgandhi/st/CVE-2010-1773.pdf.

[17] The MITRE Corporation. CVE-2010-2304. http://cve.mitre.org/cgibin/cvename.cgi?name=CVE-2010-2304.

[18] Debian Bug report logs - #586547. Webkit: CVE-2010-2304 memory corruption in rendering of list markers. https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=586547.

[19] Chromium. Diff of /branches/WebKit/375/WebCore/rendering/RenderListMarker.cpp. http://src.chromium.org/viewvc/chrome/branches/WebKit/375/WebCore/rendering/RenderListMarker.cpp?r1=48100&r2=48099.

[20] Chromium. Contents of /branches/WebKit/375/WebCore/rendering/RenderListMarker.cpp. http://src.chromium.org/viewvc/chrome/branches/WebKit/375/WebCore/rendering/RenderListMarker.cpp?annotate=48100#l104.

[21] Red Hat Bugzilla – Bug 596500 CVE-2010-1773 WebKit: off-by-one memory read out of bounds vulnerability in handling of HTML lists. https://bugzilla.redhat.com/show_bug.cgi?id=596500.

[22] B. A. Calloni, D. Campara, and N. Mansourov. White Box Definitions of Software Fault Patterns. Final Report. Lockheed Martin Corporation and KDM Analytics, Inc. 2011.

[23] R. Gandhi, H. Siy, and Y. Wu. Buffer Overflow Semantic Template. http://faculty.ist.unomaha.edu/rgandhi/st/buffer overflowtemplate.pdf.

[24] Wikipedia. Frecuency. https://en.wikipedia.org/ wiki/Frequency.

[25] Neohapsis. Yoggie Pico Pro Remote Code Execution. http://archives.neohapsis.com/archives/fulldisclosure/2007-07/0020.html.

[26] The MITRE Corporation. Common Weakness Enumeration. CWE 78. https://cwe.mitre.org/data/definitions/78.html.

[27] Cxsecurity. WebClassifieds 2005 (Auth Bypass) SQL Injection Vulnerability. http://cxsecurity.com/issue/WLB-2009010117.

[28] Mozilla Developer Network. SQL Injections. https://developer.mozilla.org/en-US/docs/Glossary/SQL_Injection.

[29] Secunia. Merak Mail Server Web Mail "IMG" HTML Tag Script Insertion. http://secunia.com/advisories/32770.

[30] N. Vijatov. Vulnerability in Merak Mail. http://secunia.com/advisories/32770.

[31] Securyty Focus. Merak Mail Server and Webmail Email Message HTML Injection Vulnerability. http://www.securityfocus.com/bid/32969/info.

[32] R. Gandhi, H. Siy, and Y. Wu. Injection Semantic Template. http://faculty.ist.unomaha.edu/rgandhi/st/injec tiontemplate.pdf.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

INFERRING PREVIOUSLY UNINSTALLED APPLICATIONS FROM DIGITAL TRACES

Jim Jones†, Tahir Khan†, Kathy Laskey†, Alex Nelson‡, Mary Laamanen‡, Doug White‡

†George Mason University, Fairfax, Virginia, United States
‡National Institute of Standards and Technology, Gaithersburg, Maryland, United States

ABSTRACT

In this paper, we present an approach and experimental results to suggest the past presence of an application after the application has been uninstalled and the system has remained in use. Current techniques rely on the recovery of intact artifacts and traces, e.g., whole files, Windows Registry entries, or log file entries, while our approach requires no intact artifact recovery and leverages trace evidence in the form of residual partial files. In the case of recently uninstalled applications or an instrumented infrastructure, artifacts and traces may be intact and complete. In most cases, however, digital artifacts and traces are altered, destroyed, and disassociated over time due to normal system operation and deliberate obfuscation activity. As a result, analysts are often presented with partial and incomplete artifacts and traces from which defensible conclusions must be drawn. In this work, we match the sectors from a hard disk of interest to a previously constructed catalog of full files captured while various applications were installed, used, and uninstalled. The sectors composing the files in the catalog are not necessarily unique to each file or application, so we use an inverse frequency-weighting scheme to compute the inferential value of matched sectors. Similarly, we compute the fraction of full files associated with each application that is matched, where each file with a sector match is weighted by the fraction of total catalog sectors matched for that file. We compared results using both the sector-weighted and file-weighted values for known ground truth test images and final snapshot images from the M57 Patents Scenario data set. The file-weighted measure was slightly more accurate than the sector-weighted measure, although both identified all of the uninstalled applications in the test images and a high percentage of installed and uninstalled applications in the M57 data set, with minimal false positives for both sets. The key contribution of our work is the suggestion of uninstalled applications through weighted measurement of residual file fragments. Our experimental results indicate that past application activity can be reliably indicated even after an application has been uninstalled and the host system has been rebooted and used. The rapid and reliable indication of previously uninstalled applications is useful for cyber defense, law enforcement, and intelligence operations.

Keywords: digital forensics; digital artifact; digital trace; partial artifact; residual artifact; uninstalled application

Jones, Jim; Kahn, Tahir; Laamanen, Mary; Laskey, Kathryn; Nelson, Alexander; White, Douglas.
"Inferring previously uninstalled applications from digital traces."
Paper presented at 11th Annual ADFSL Conference on Digital Forensics, Security and Law, Daytona Beach, FL. May 24, 2016 - May 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

1. INTRODUCTION

The practice of digital forensics is the art and science of inferring and proving past activity given some set of residual digital artifacts and traces. These artifacts and traces may be files, Windows Registry entries, log entries, memory contents, network traffic, etc., and past activity of interest may be legitimate and illegitimate user activity, system activity, application installation and usage, malware infection and operation, etc. While whole artifacts may be recoverable in some cases, many situations require inferring and proving past activity given residual partial artifacts and traces. We propose that past activity, specifically application installation and usage, can be reliably suggested from digital traces, even when the application in question has been uninstalled and usage of the system and media has continued. We assume that full artifacts created by an activity degrade monotonically and non-linearly over time. Specifically, files created as a consequence of application installation, usage, and uninstallation are subsequently deleted, and some sectors from these deleted files will be overwritten while other sectors may persist on the digital media. Given prior knowledge of the full file artifacts created by an application, we can then search media of interest for traces in the form of matching partial artifacts, i.e., sectors from the original full artifact, and reason over these matches to suggest past application presence. Our approach complements existing methods that rely on evidence from intact full-file artifacts, an uncleansed Windows Registry, intact log entries., or traces from other sources such as memory contents or network traffic.

In the sections that follow, we discuss prior work in this area, then we describe the two core elements of our approach: (i) building a catalog of sectors associated with specific applications, and (ii) reasoning over sectors that match entries in that catalog. Subsequent sections present our experimental results against a test set with known ground truth and the M57 Patents Scenario (Woods et al., 2011) disk images, for which we have some ground truth. We close the paper with a summary of our conclusions, limitations of this approach, and future research plans.

2. RELATED WORK

Related work to establish the presence of installed and uninstalled applications has generally relied on intact file artifacts (Koppen et al., 2013; Quick et al., 2013), log file analysis (Forte, 2004), and examination of the Windows registry when available (Laamanen et al., 2014; Nelson et al., 2014; Wong, 2007). Additional techniques and methods rely on traces such as email addresses, URLs, etc. extracted from raw data (Garfinkel, 2013), or data structures and other known-layout data from memory (Ligh et al., 2014). Intact file artifacts for uninstalled applications may be files remaining from an aborted or poorly written uninstall application, or may be user files which are created during application use and are deliberately not deleted as part of the application uninstall process, such as user preference files. Log files include varying levels of detail depending on the application creating the log, and establishing the integrity of the log file requires secure creation, transmission, and storage of the log file. Registry artifacts may include application specific keys as well as command line execution arguments, recently accessed files, and similar indicators of application installation and usage, whether the application in question has been uninstalled or not. In contrast, our work does not require recovery of any intact artifacts and is specifically designed to suggest applications that have been uninstalled.

Our work relies on recovery and analysis of file fragments in the form of disk sectors. Collange, Dandass, Daumas, and Defour (Collange et al., 2009), Garfinkel, Nelson, White, and Roussev (Garfinkel et al., 2010), and later Young, Foster, Garfinkel, and Fairbanks (Young et al., 2012) and Foster (Foster, 2012) examined sector content uniqueness as it relates to specific file identification. This initial work successfully identified files with distinct content, such as videos, from a limited number of sectors, but the later work also hinted at issues with sector content common across multiple files. These issues fully emerged in the work of Garfinkel and McCarrin (Garfinkel et al., 2015) in the form of "common data structures found in Microsoft Office documents and multimedia files." Garfinkel and McCarrin label such

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

file fragments "non-probative blocks" and developed heuristics to account for these blocks and reliably detect file presence from fragments. By comparison, we are inferring the past presence of applications based on blocks from multiple files. Further, our approach pre-selects potentially probative blocks then weights matching blocks based on their frequency in our catalog.

3. APPROACH AND METHODOLOGY

The theory underpinning our approach is that application installation and use creates files, and application uninstallation deletes these files. The sectors containing the contents of these deleted files are overwritten over time, but some sectors may remain intact until subsequent examination. These residual sectors, or traces, may be used to infer the likelihood that a particular application was previously installed on the examined system.

It is important to note that just because we empirically establish that an application installation and use creates a specific set of files and corresponding sectors, this does not imply that the presence of these sectors or even intact files proves the current or past presence of the application in question. That is, if I know A causes B and I subsequently find B, I cannot logically conclude that A occurred. On the other hand, if A is established to be the only possible cause of B, then I can logically conclude that the presence of B does prove A. In the context of files and associated sectors, prior research (Garfinkel et al., 2010)(Garfinkel and McCarrin, 2015) showed that while a sector may not have only one possible producer, in practice it is likely to have only one, especially for high entropy sectors. In our work, a pre-processing step removes sectors appearing in our clean OS images, sectors with low entropy, and sectors appearing more than 100 times in our initial catalog, thereby removing sectors known to be produced by, or likely produced by, other processes. Further, we weight the influence of sectors based on the number of different catalog applications in which they appear. In practice, this is accomplished by our Inverse Document Frequency weight described below. Finally, we note that we are not proving the past presence of an application. Rather, we are suggesting an increased likelihood that a particular application was present at some past time, where proof to the standard required by the circumstances would have to be obtained from additional evidence.

Our approach, summarized in Figure 1, reasons over media sectors that match entries in a catalog associating sectors with specific application activities. The catalog was created for 16 Windows applications in a controlled environment using virtual machine snapshots. Catalog entries are post-processed to remove less useful sectors and to assess each sector's potential inferential value. We then match sectors from a digital storage device of interest, e.g., a hard drive, to the entries in the catalog and compute weighted measures that represent the likelihood that the associated application was previously installed on the media of interest.

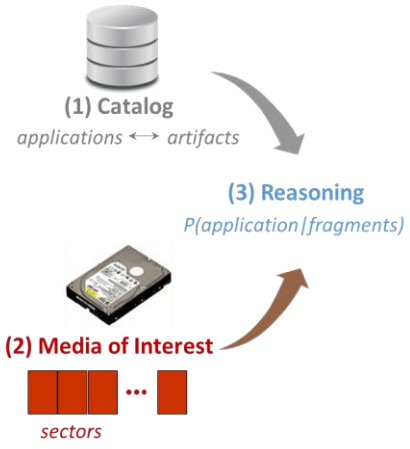


Figure 1: Approach Overview

3.1. Catalog Creation and Post-processing

We are leveraging the NIST Diskprinting effort (Laamanen et al., 2014) to collect application traces. Diskprinting uses virtual machine snapshots to record the state of a system before and after an action of interest. Each snapshot together with captured network traffic is called a *slice*. A series of slices, which reflect sequential activities regarding a single application, is called a *diskprint*. The contents of two adjacent snapshots may then be compared to extract differences (Figure 2). For our purposes, the file systems of adjacent snapshots are compared to identify new, modified, or deleted files. For the NIST diskprinting data, these activities are application Install, Open, Close, Uninstall, and system Reboot

(indicated as I, O, C, U, and R in Figure 3). Diskprints are made up of sequential and cumulative slices, hence the nomenclature B (Base), BI (Base + Install), ..., BIOCUR (Base + Install + Open + Close + Uninstall + Reboot) in Figure 3. Diskprints are created with shared baseline states, by rolling the virtual machine state back to a common point before applications were installed, in order to isolate effects of the operating system.

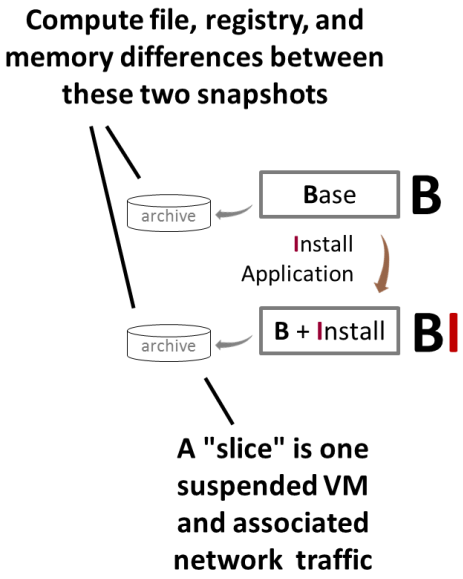


Figure 2: Slices and differencing

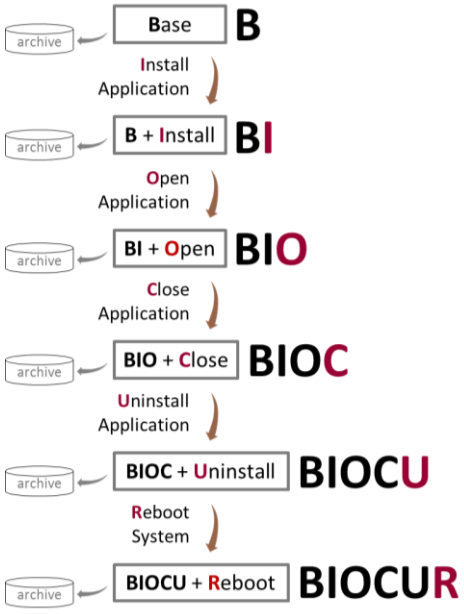


Figure 3: A diskprint is a series of related slices

We use 29 application diskprints of the NIST diskprint data (NIST, 2015), representing 16 applications across one or more different Windows platforms (Table 1) plus three clean Operating System diskprints: one WinXP and two Win7. The applications were selected in part to facilitate subsequent testing against the M57 Patents Scenario images.

Table 1: NIST diskprints

	WinXP	Win7x32	Win7x64
Adv Keylogger	✓		
Chrome	✓	✓	✓
Eraser		✓	
Firefox	✓	✓	✓
HxD hex editor		✓	
Invisible Secrets	✓		
MS Office	✓	✓	✓
Python	✓		
Safari	✓	✓	✓
Sdelete		✓	✓
Thunderbird	✓		
TrueCrypt	✓		
UPX		✓	✓
WinRar		✓	✓
WinZip		✓	✓
Wireshark		✓	✓

Each NIST Diskprint slice contains a snapshot of the system hard disk in the form of a VMDK file. For each pair of adjacent slices, we computed file differences and 512-byte sector-aligned MD5 hashes for each new or modified file (Garfinkel et al., 2012). For experimental purposes, we used MD5s because of their smaller bit count and acceptable impact of false positives from MD5 weaknesses (Dandass et al., 2008). However, an operational deployment of this research would need to employ a more secure cryptographic hash per NIST guidelines on hash selection (NIST, 2012). The diskprint sector hash data currently computes the final sector hash of each file based on file extant vs. padding the final sub-sector fragment with zeros and computing a 512-byte hash. We discard these sub-512 byte sector hashes since they will never match our media of interest hashes, which are always based on a full 512-byte sector hash. We have

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

discussed but not implemented padding sub-512 byte diskprint fragments with zeros prior to computing the MD5 hash. We process the diskprint sector hash data as described in the following paragraphs and ingest the data into a hashdb (NPS-DEEP, 2015) instance.

File differencing as implemented on the diskprint data has the potential to capture spurious traces, i.e., file differences that are not related to the activity in question but are the product of unrelated system activity. We describe this property of a file as *attribution*, where *positive attribution* means a file is a result of the activity in question, *negative attribution* means the file is not the result of the activity in question, and *marginal attribution* means the file is due to the activity in question but in a non-probative way (such as the $BitMap or pagefile.sys files on a Windows system).

Positive attribution is determined by keyword searching of the filename and path associated with each sector hash. This information is stored in files using differentially-annotated Digital Forensics XML, or DFXML (Garfinkel, 2012; Nelson et al., 2014)), a language that associates file system metadata with file content summaries, including file paths, full-file hashes, and sector-level hashes. The DFXML language facilitates interaction between tools, such as those used in our processing steps.

For each application, sector hashes whose source file paths contain matching keywords from Table 2 are retained. Keywords were derived by examining string frequencies in the collective file path names for each application and selecting the most common, subject to human review.

Table 2: Keyword Whitelist

Application	keyword(s)
Adv Keylogger	keylogger
Chrome	chrome,google
Eraser	eraser
Firefox	firefox,mozilla
HxD hex editor	hxd
Invisible Secrets	"invisible secrets"
MS Office	office,"microsoft shared"
Python	python
Safari	safari
Sdelete	sdelete
Thunderbird	thunderbird
TrueCrypt	truecrypt
UPX	upx
WinRar	winrar
WinZip	winzip
Wireshark	wireshark

For example, whitelisting the Firefox19 on 64-bit Windows 7 (Win7x64) diskprint reduced trace files from 1,054 to 289, and reduced associated sector hashes from 16,096,960 to 157,530. This whitelisting approach is something of a blunt instrument, yet we obtain good results in our subsequent experiments. In the section on future work, we propose alternative catalog construction techniques to increase the quality of collected file fragments (sectors).

Sector hashes, including those from files with positive attribution, are not necessarily unique. We describe this property of a sector as its *frequency*, where *distinct* means the sector only occurs once in the post-processed diskprint data, *application common* means the sector occurs in one or more application diskprints but not elsewhere, and *global* means the sector occurs outside of the application diskprints (i.e., in the baseline OS states).

We limit sector hash value frequency in the hashdb instance to 100. While somewhat arbitrary, this limit allows for some multi-application or multi-print hashes to remain while removing hashes not likely to have discriminatory value and keeping the hashdb to a manageable size. If desired, we can later select hash values below the f=100 threshold, or we can reprocess the original diskprint sector hash data if results indicate that sector hashes with frequency greater than 100 have inferential value.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Table 3: Total hashes and files per application diskprint

Diskprint	Total Hashes	Total Files
AdvKeylogger-WinXP	4,716	23
Chrome28-W7x32	686,986	669
Chrome28-W7x64	670,051	499
Chrome28-WinXP	1,035,098	624
eraser-W7x32	69,984	24
Firefox19-W7x32	103,341	132
Firefox19-W7x64	106,270	146
Firefox19-WinXP	96,377	115
HxD171-W7x32	4,774	12
InvSecrets21-WinXP	6,689	19
OfficePro2003-W7x32	1,090,216	3,800
OfficePro2003-W7x64	1,077,126	3,804
OfficePro2003-WinXP	656,354	2,801
Python264-WinXP	86,287	2,355
Safari157-W7x32	316,224	907
Safari157-W7x64	569,645	1,504
Safari157-WinXP	343,824	918
sdelete-W7x32	642	5
sdelete-W7x64	642	4
Thunderbird2-WinXP	68,102	172
TrueCrypt63-WinXP	24,520	16
UPX-W7x32	1,796	19
UPX-W7x64	1,813	19
Winrar5beta-W7x32	9,196	41
Winrar5beta-W7x64	18,328	81
Winzip17pro-W7x32	240,229	149
Winzip17pro-W7x64	262,854	153
Wireshark-W7x32	171,515	617
Wireshark-W7x64	209,666	611
TOTALS	7,933,265	20,239

As a practical matter, hashdb supports a maximum frequency parameter when the hashdb instance is created. However, this only prevents the addition of more hash values which have already reached the maximum frequency - it does not remove the hash value from the hashdb instance. To prevent undesired

effects on our subsequent calculations, we set a maximum frequency of 101 prior to ingest, then we remove all hashes with frequency of 101 after ingest is complete. Without this extra step, the catalog would contain all sector hashes in the original data and all sector hashes with frequency greater than 100 would be retained with frequency equal to 101, regardless of the actual frequency of these sector hashes. With this extra step, the catalog contains sector hashes with accurate frequency counts, and only sector hashes with actual frequencies of 100 or less.

Certain low entropy sector contents, such as all zeros or all ones, occur with high frequency in many sources and have no discriminatory value. For example, the following sector hashes with the noted content occur thousands or millions of times in the original NIST diskprint sector hash data.

```
'bf619eac0cdf3f68d496ea9344137e8b' # repeated 00
'393a0fa0f348fb03871ab93726057ddc' # repeated 01
'de03fe65a6765caa8c91343acc62cffc' # repeated FF
'c5d77850e62433f25d5496bfad94c1b2' # repeated 00; 06 @offset 510
```

These sector hashes would be removed by our maximum frequency processing step above. However, we filter these sector hashes out at an earlier processing stage simply to speed up subsequent processing.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The NIST diskprint data includes diskprints of non-application activities on three operating system variants: two WinXP and one Win7x64. Any hash value appearing in these base OS diskprints does not have discriminatory value for a subsequently installed application, so we remove these hash values from the hashdb instance. As a practical matter, this was accomplished by using hashdb's *add_repository* command to build a hashdb instance of all OS hash values, then using hashdb's *subtract_hash* command to remove those hash values from the original hashdb instance.

The combined whitelist and frequency limit processing resulted in an overall file count of 99,227 and sector hash count of 44,677,825 (file and sector hash counts before whitelist and frequency limits were not computed). Removing the base OS diskprint sector hash values reduced the overall file count to 20,239 and sector hash count to 7,933,265.

To facilitate later calculations of application likelihood, we count and store the total sector hashes and total files per application in the final catalog. These totals (Table 3) are extracted from the final noise-reduced hashdb instance using hashdb's *hash_table* command (v.1.0.0 and prior) with subsequent grep expressions (for hash totals) and hashdb's *sources* command with subsequent grep expressions (for file totals). These totals are for all slices in each diskprint, where each diskprint contains 5-6 slices. 5-slice diskprints result from applications where the "Open" and "Close" steps were combined as a single "Run" step.

3.2. Image Processing

Media of interest is assumed to be a raw image of a hard disk or similar. We use the md5deep tool (https://github.com/jessek/hashdeep) to compute sector-aligned 512-byte MD5 hashes for the entire disk or disk image, storing the results in a DFXML file. We then use hashdb's *scan_expanded* command (v1.0.0) to identify hash values in the DFXML file that match hash values from the hashdb instance. The hashdb *scan_expanded* command output includes the file source and repository information from the hashdb instance. We require these details, as we are using the repository name to hold the diskprint (application) identifier, and the source file information allows us to compute which files in the catalog, and how much of each file, is matched. Matches are written to an interim text file.

The matches text file is processed to compute the various measures of diskprint matching, i.e., application presence. Output includes the number and fraction of distinct hashes matched for each diskprint (application), the number and fraction of total files matched for each diskprint where a file match is declared if one or more hash values from that file are matched. Output also includes weighted versions of these two measures, which are discussed below.

After we eliminated weak or non-probative sector hash values in our noise reduction process, we then applied weights to matching sector hashes based on their occurrence across applications, i.e., their frequency in the catalog. A sector hash that occurs in N different diskprinted applications is weighted with a factor 1/N (this is the hyperbolic formulation of Inverse Document Frequency described by Zobel and Moffat (Zobel et al., 1998)). A sector hash value that occurs in only one diskprinted application is weighted 1/1=1.0; a sector hash value that occurs in 2 different diskprinted applications is weighted 1/2=0.5; and so on. This calculation is shown below, and the results are shown in the sample output of Table 4 under the heading w_sector% (weighted sector %). In the formula, each matching sector for a given diskprint is weighted by its inverse frequency in the catalog; these weighted matching sector counts are then summed and divided by the total number of sectors in the catalog for that diskprint to give a weighted sector % for that diskprint.

$$\text{weighted sector \%}_{DP} = \left(\sum_{S=1}^{\text{num\_sec\_matches}} 1 / \text{freq}_S \right) / \text{sectors\_total}_{DP}$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Instead of declaring a file *present* if one or more hash values from that file are found (as the data in Table 4 under the heading "files_found" does), we compute the percent of each file that is matched and weight the summation accordingly. For example, if we match M sectors for a file out of N total sectors in the catalog for that file, then that file hit is worth M/N. We sum all of the weighted file hits for each diskprint and divide by the total number of files in the catalog for that diskprint to give a weighted file % for that diskprint. This calculation is shown below, and the results are shown in the sample output of Figure 4 under the heading w_file% (weighted file %).

$$\text{weighted file \%}_{DP} = \left(\sum_{F=1}^{\text{num\_file\_matches}} \frac{\text{matched\_sectors}_F}{\text{total\_sectors}_F} \right) / \text{files\_total}_{DP}$$

Sample output for one of the test images is shown in Table 4 below. This Win7x64 test image had Chrome installed, opened, closed, and uninstalled, then the system was rebooted and the snapshot taken. The three Chrome diskprints (for Chrome on WinXP, Win7x32, and Win7x64) are the three highest valued hits based on both weighted sector % and weighted file % (the sort key in the table). Other data are included in this verbose output, to include the total sector hashes and total files for each diskprint, as well as hits and % of total for each.

Table 4: Sample analysis output for source image "Chrome Win7x64"

diskprintName	sectors found	sectors total	sector%	w_sector%	files found	files total	file%	w_file%
Chrome28-W7x64	66795	670051	9.97%	3.63%	153	499	30.66%	21.46%
Chrome28-WinXP	40831	1035098	3.94%	1.16%	208	624	33.33%	21.10%
Chrome28-W7x32	66795	686986	9.72%	3.54%	152	669	22.72%	16.26%
Winzip17pro-W7x32	2186	240229	0.91%	0.46%	41	149	27.52%	3.63%
Winzip17pro-W7x64	2162	262854	0.82%	0.41%	42	153	27.45%	3.53%
Firefox19-W7x32	4183	103341	4.05%	0.59%	18	132	13.64%	2.44%
Firefox19-WinXP	4183	96377	4.34%	0.63%	17	115	14.78%	2.40%
Firefox19-W7x64	4184	106270	3.94%	0.57%	18	146	12.33%	2.37%
Thunderbird2-WinXP	17	68102	0.02%	0.01%	6	172	3.49%	1.09%
Winrar5beta-W7x64	9	18328	0.05%	0.01%	7	81	8.64%	0.38%
Winrar5beta-W7x32	9	9196	0.10%	0.02%	7	41	17.07%	0.38%
Safari157-WinXP	573	343824	0.17%	0.02%	31	918	3.38%	0.32%
Safari157-W7x32	573	316224	0.18%	0.02%	31	907	3.42%	0.30%
Safari157-W7x64	575	569645	0.10%	0.01%	35	1504	2.33%	0.24%
sdelete-W7x64	1	642	0.16%	0.04%	2	4	50.00%	0.17%
Wireshark-W7x32	51	171515	0.03%	0.01%	10	617	1.62%	0.16%
sdelete-W7x32	1	642	0.16%	0.04%	2	5	40.00%	0.14%
OfficePro2003-WinXP	1014	656354	0.15%	0.02%	33	2801	1.18%	0.13%
OfficePro2003-W7x32	1014	1090216	0.09%	0.01%	33	3800	0.87%	0.11%
OfficePro2003-W7x64	1014	1077126	0.09%	0.01%	33	3804	0.87%	0.08%
Wireshark-W7x64	11	209666	0.01%	0.00%	5	611	0.82%	0.02%
eraser-W7x32	21	69984	0.03%	0.02%	2	24	8.33%	0.01%
TrueCrypt63-WinXP	1	24520	0.00%	0.00%	1	16	6.25%	0.01%
Python264-WinXP	23	86287	0.03%	0.01%	6	2355	0.25%	0.00%
AdvKeylogger-WinXP	0	4716	0.00%	0.00%	0	23	0.00%	0.00%
InvSecrets21-WinXP	0	6689	0.00%	0.00%	0	19	0.00%	0.00%
UPX-W7x32	0	1796	0.00%	0.00%	0	19	0.00%	0.00%
HxD171-W7x32	0	4774	0.00%	0.00%	0	12	0.00%	0.00%
UPX-W7x64	0	1813	0.00%	0.00%	0	19	0.00%	0.00%

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

4. RESULTS

We generated eight test images, five containing the installation, use, and uninstallation of a single catalog application and three containing the installation, use, and uninstallation of multiple catalog applications. We also processed the four final day disk images from the M57 Patents Scenario case. We also processed WinXP, Win7x32, and Win7x64 images with no applications of interest installed and found no more than 1% matching sectors per application.

Table 5: Single application test case results

Source Image: Chrome28-W7x64			**Source Image: UPX-W7x64**		
diskprintName	w_sector%	w_file%	diskprintName	w_sector%	w_file%
Chrome28-W7x64	3.63%	21.46%	UPX-W7x32	2.97%	52.16%
Chrome28-WinXP	1.16%	21.10%	UPX-W7x64	2.94%	52.16%
Chrome28-W7x32	3.54%	16.26%	Winzip17pro-W7x32	0.44%	3.52%
Winzip17pro-W7x32	0.46%	3.63%	Winzip17pro-W7x64	0.41%	3.45%
Winzip17pro-W7x64	0.41%	3.53%	Firefox19-W7x64	0.01%	1.69%
...	...	...	...	...	...
Source Image: Winrar5beta-W7x64			**Source Image: Firefox19-W7x64**		
diskprintName	w_sector%	w_file%	diskprintName	w_sector%	w_file%
Winrar5beta-W7x32	8.39%	56.18%	Firefox19-WinXP	6.88%	57.32%
Winrar5beta-W7x64	4.21%	32.80%	Firefox19-W7x32	6.42%	51.52%
Winzip17pro-W7x32	0.44%	3.53%	Firefox19-W7x64	6.26%	47.25%
Winzip17pro-W7x64	0.41%	3.46%	Winzip17pro-W7x32	0.44%	3.57%
sdelete-W7x32	0.04%	0.14%	Winzip17pro-W7x64	0.41%	3.48%
...	...	...	...	...	...
Source Image: sdelete-W7x64					
diskprintName	w_sector%	w_file%			
sdelete-W7x64	7.75%	33.95%			
sdelete-W7x32	7.75%	27.16%			
Winzip17pro-W7x32	0.44%	3.52%			
Winzip17pro-W7x64	0.41%	3.45%			
Firefox19-W7x64	0.01%	1.67%			
...	...	...			

4.1. Single-application test cases

For each single application test case, we started with a fresh install of the appropriate OS (WinXP, Win7x32, or Win7x64) and mimicked the diskprint activity as described in the diskprint data, e.g., install, open, close, uninstall, and reboot. These test cases did not use NIST's source media for the OS or application, and did not strictly follow the details of activity performed by NIST personnel when creating the diskprint images. Results from the post-reboot snapshot of the seven single application test cases are summarized in Table 5, where only the top 5 weighted file % results are shown. In each test case, the installed/uninstalled application was correctly identified and the weighted sector % and weighted file % measures indicate a sharp drop off for catalog applications that were not present on that test image.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

4.2. Multiple-application test cases

Table 6: Multiple application test case results

Source Image: Firefox, Chrome, & Safari		**Source Image: WinRAR & WinZip**	
diskprintName	w_file%	diskprintName	w_file%
Safari157-W7x32	94.32%	Winzip17pro-W7x64	35.60%
Safari157-WinXP	92.80%	Winzip17pro-W7x32	34.88%
Safari157-W7x64	57.12%	Winrar5beta-W7x32	9.97%
Firefox19-WinXP	46.57%	Winrar5beta-W7x64	9.29%
Firefox19-W7x32	42.83%	Firefox19-WinXP	2.66%
Firefox19-W7x64	37.73%	Firefox19-W7x64	2.60%
Chrome28-WinXP	22.10%	Firefox19-W7x32	2.23%
Chrome28-W7x64	12.88%	Thunderbird2-WinXP	1.49%
Chrome28-W7x32	9.84%	sdelete-W7x64	0.17%
Winzip17pro-W7x32	3.62%	Wireshark-W7x32	0.16%
...	...	...	...
Source Image: Chrome & Firefox			
diskprintName	w_file%		
Firefox19-WinXP	57.21%		
Firefox19-W7x32	52.04%		
Firefox19-W7x64	47.33%		
Chrome28-W7x64	20.45%		
Chrome28-WinXP	20.37%		
Chrome28-W7x32	15.58%		
Winzip17pro-W7x32	3.64%		
Winzip17pro-W7x64	3.53%		
Thunderbird2-WinXP	1.68%		
Winrar5beta-W7x64	0.42%		
...	...		

Three test cases were constructed in a manner similar to the single application test cases, but multiple applications were installed, used, and uninstalled, and multiple reboots occurred. Two of these test cases incorporated two applications and one incorporated three applications. Results from the post-reboot snapshot of these three multiple application test cases are summarized in Table 6. For these cases, the top 10 results based on weighted file % are shown. In all three cases, all installed/uninstalled applications are correctly identified, after which the weighted file % drops off sharply.

4.3. M57 Patents Scenario images

The M57 Patents Scenario is a publicly available data set. The scenario was created for educational and research purposes by faculty and students at the Naval Postgraduate School. The creators of the data set mimicked criminal activity in a lab environment over the course of a month, capturing disk and device images and network traffic during the exercise. Scenario documentation includes a description of the systems and networks involved, characters, and a storyline. For our purposes, the final day snapshots are sufficiently realistic system images, by merit of having been physical machines operated for a real-world month. Also, we have some ground truth about installed and uninstalled applications based on the scenario documentation (availability restricted to faculty at accredited institutions), work by Roussev & Quates (Roussev et al., 2012) that analyzed the same images, and our own direct analysis of the scenario

images. Results from processing the final day (2009-12-11) images for the four scenario users (Charlie, Jo, Pat, and Terry) are summarized in Table 7, where the host OS is indicated after the system name.

Table 7: M57 Patents Scenario results

Charlie (XP)		Jo (XP)		Pat (XP)		Terry (Vista)	
diskprintName	w_file%	diskprintName	w_file%	diskprintName	w_file%	diskprintName	w_file%
Python264-WinXP	98.98%	Python264-WinXP	98.83%	Python264-WinXP	98.91%	Python264-WinXP	85.52%
InvSecrets21-WinXP	63.16%	TrueCrypt63-WinXP	50.00%	Thunderbird2-WinXP	24.94%	Thunderbird2-WinXP	27.81%
Thunderbird2-WinXP	61.00%	Thunderbird2-WinXP	24.73%	AdvKeylogger-WinXP	21.97%	Winzip17pro-W7x64	10.37%
Safari157-W7x32	10.25%	Safari157-W7x32	11.35%	HxD171-W7x32	8.39%	Winzip17pro-W7x32	10.05%
Safari157-WinXP	10.16%	Safari157-WinXP	11.26%	Firefox19-WinXP	3.17%	HxD171-W7x32	8.37%
Safari157-W7x64	6.69%	Safari157-W7x64	7.37%	Firefox19-W7x64	2.93%	Safari157-W7x32	5.46%
Firefox19-WinXP	3.26%	Firefox19-WinXP	3.24%	Firefox19-W7x32	2.78%	Safari157-WinXP	5.35%
Firefox19-W7x32	2.77%	Firefox19-W7x32	2.74%	Winzip17pro-W7x64	2.03%	Chrome28-WinXP	4.83%
Firefox19-W7x64	2.50%	Firefox19-W7x64	2.62%	Chrome28-WinXP	1.64%	Chrome28-W7x64	4.81%
Chrome28-WinXP	2.11%	Chrome28-WinXP	2.15%	Chrome28-W7x64	1.63%	Firefox19-WinXP	3.59%
Winzip17pro-W7x64	2.08%	Chrome28-W7x64	2.03%	Winzip17pro-W7x32	1.50%	Chrome28-W7x32	3.59%
Chrome28-W7x64	2.02%	Chrome28-W7x32	1.52%	Chrome28-W7x32	1.22%	Firefox19-W7x64	3.56%
Chrome28-W7x32	1.52%	sdelete-W7x64	1.35%	TrueCrypt63-WinXP	1.22%	Firefox19-W7x32	3.55%
Winzip17pro-W7x32	1.51%	Winzip17pro-W7x64	1.26%	Winrar5beta-W7x64	0.85%	Safari157-W7x64	3.47%
sdelete-W7x64	1.35%	sdelete-W7x32	1.08%	Winrar5beta-W7x32	0.84%	Winrar5beta-W7x64	2.21%
sdelete-W7x32	1.08%	Winrar5beta-W7x64	0.95%	Safari157-WinXP	0.62%	Winrar5beta-W7x32	2.19%
TrueCrypt63-WinXP	0.73%	Winrar5beta-W7x32	0.94%	Safari157-W7x32	0.54%	TrueCrypt63-WinXP	0.97%
Winrar5beta-W7x32	0.64%	Winzip17pro-W7x32	0.72%	OfficePro2003-WinXP	0.47%	OfficePro2003-W7x32	0.39%
Winrar5beta-W7x64	0.64%	OfficePro2003-WinXP	0.43%	OfficePro2003-W7x32	0.45%	OfficePro2003-WinXP	0.35%
OfficePro2003-WinXP	0.37%	OfficePro2003-W7x32	0.41%	OfficePro2003-W7x64	0.42%	OfficePro2003-W7x64	0.35%
OfficePro2003-W7x32	0.32%	OfficePro2003-W7x64	0.37%	Safari157-W7x64	0.39%	Wireshark-W7x32	0.09%
OfficePro2003-W7x64	0.31%	Wireshark-W7x32	0.07%	Wireshark-W7x32	0.10%	eraser-W7x32	0.05%
Wireshark-W7x32	0.06%	HxD171-W7x32	0.04%	Wireshark-W7x64	0.02%	Wireshark-W7x64	0.05%
eraser-W7x32	0.01%	eraser-W7x32	0.02%	eraser-W7x32	0.02%	AdvKeylogger-WinXP	0.03%
AdvKeylogger-WinXP	0.01%	Wireshark-W7x64	0.02%	InvSecrets21-WinXP	0.00%	InvSecrets21-WinXP	0.00%
Wireshark-W7x64	0.00%	AdvKeylogger-WinXP	0.01%	UPX-W7x32	0.00%	UPX-W7x32	0.00%
UPX-W7x32	0.00%	InvSecrets21-WinXP	0.00%	sdelete-W7x32	0.00%	sdelete-W7x32	0.00%
HxD171-W7x32	0.00%	UPX-W7x32	0.00%	UPX-W7x64	0.00%	UPX-W7x64	0.00%
UPX-W7x64	0.00%	UPX-W7x64	0.00%	sdelete-W7x64	0.00%	sdelete-W7x64	0.00%

Legend	True positive	True negative	False positive	False negative	Different OS

In the M57 results of Table 7, green cells indicate true positives, which are confirmed installed or uninstalled programs based on the scenario documentation, other published analysis, and direct forensic examination of the scenario images. White cells are true negatives, similarly verified. Red cells indicate false positives, which we define as weighted file % scores above the lowest true positive. Blue cells are false negatives, which we define as a known installed applications with a weighted file % lower than at least one true negative. Yellow cells indicate other OS versions of detected applications. For the true positives, Python and Firefox installations are confirmed for all four systems. For the Charlie system, Thunderbird is also confirmed by the scenario documentation, and Invisible Secrets is suggested by the scenario documentation ("...emails proprietary information steganographically hidden in JPEG image...") and confirmed by Roussev and Quates as well as a direct examination of the image. The presence of TrueCrypt on the Jo system, Advanced Keylogger on the Pat system, and Chrome on the Terry system are all confirmed in the scenario documentation. Advanced Keylogger is also confirmed in the scenario documentation to have been uninstalled prior to the Pat 2009-12-11 image.

We examined the scenario images directly using Autopsy 4.0.0 in an effort to understand the apparent false positives and the lone false negative (eraser on the Terry image). A summary of our preliminary

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

findings is below in Table 8. A more extensive analysis is underway to establish if these are in fact false positives, or if some of them represent as yet undocumented true positives. The results of this analysis will be reported in future work.

Table 8: False positive and false negative preliminary analysis

(System(s)) **Application**	Preliminary Analysis
(Charlie/Jo/Terry) **Safari**	Apple's QuickTime and Apple's software update applications are present on the Charlie and Jo systems and may explain the Safari results due to catalog artifacts in common (Safari would include the Apple software update application and possibly QuickTime as well). The Terry system also indicated Safari, although at a lower level than the Charlie and Jo systems, but the Terry system does not show indications of a QuickTime installation.
(Jo/Pat/Terry) **Thunderbird**	Thunderbird is known to have been installed on the Charlie system on 11-12-2009, but is not documented or apparent on the other three systems. It is possible that Thunderbird was installed on all four systems on 11-12-2009 but immediately uninstalled on the three non-Charlie systems.
(Pat/Terry) **HxD**	HxD may have been installed and uninstalled between snapshots, hence no entries were found in locations like Program Files. The Cygnus hex editor was confirmed on the Charlie system, so the scenario operators are know to have installed a hex editor, although a different one than HxD detected on the Pat and Terry systems.
(Terry) **Winzip**	Possibly due to compression libraries bundled in Windows Vista and also used by Winzip, but not bundled in Windows XP.
(Terry) **Eraser**	Likely due to a difference in application versions between the catalog and the M57 image. Most of the eraser sectors in the catalog come from the eraser.exe file, hence a minor change in the compiled code would prevent sector matches. The eraser application has a small number of files, hence is more susceptible to such a variation than other applications with large numbers of files and hence unchanged sectors across versions.

Of particular interest in the M57 results is the successful detection of Advanced Keylogger on the Pat system after uninstallation and continued use. Such detections are the main goal of our work and is distinct from other work such as Roussev and Quates that relied on mid-scenario snapshots to detect Advanced Keylogger. In contrast, our approach detected Advanced Keylogger using only the final scenario snapshot, after Advanced Keylogger had been uninstalled and the system used for five additional days. Figure 4 shows the presence and persistence of Advanced Keylogger sector artifacts over the life of the scenario. The data consists of 17 snapshots over 26 calendar days, where days without snapshots are indicated by an asterisk along the X-axis of Figure 4. The vertical axis in the graph, sector %, is the matched sectors as a fraction of the total sectors associated with Advanced Keylogger in the catalog. Advanced Keylogger was installed between the 12/2 and 12/3 snapshots, and uninstalled between the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

12/4 and 12/7 snapshots. Subsequent system usage further destroyed probative sectors, yet our weighted file % measure still detected Advanced Keylogger in the 12/11 snapshot (21.97% based on the remaining 24 sectors from 8 different files). We speculate that 100% of the catalog sectors were not matched in the 12/3 and 12/4 snapshots due to slight differences in artifacts created during installation and use of Advanced Keylogger the different systems of the catalog and the M57 scenario.

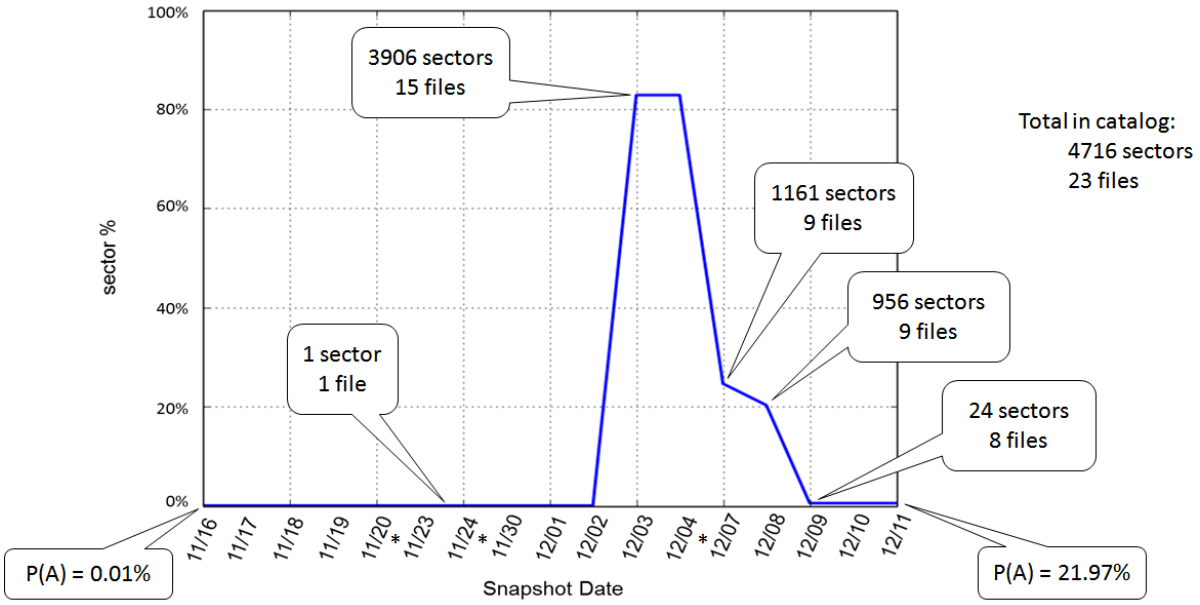


Figure 4: Sector artifact persistence for Advanced Keylogger on Pat's M57 system

One unresolved issue is to determine the threshold at which an application should be considered present or previously present. While the M57 results might indicate a weighted file % threshold of about 3%, the contents of deleted files are modified (destroyed) over time, so a single threshold for uninstalled applications is unlikely to exist. However, we are conducting related work to model the persistence of deleted files over time under different artifact and system usage conditions. This related works aims to provide a basis for asserting the implications of a particular weighted file % for a specific application after a known amount of time and activity. Additionally, our catalog of 16 applications tested on the four images of the M57 data set is not large enough to conclude statistical significance. However, as a practical matter, the current use cases for our approach are (a) for an analyst to work down the list in decreasing weighted file % score until applications are no longer confirmed or no longer of interest, or (b) to have a specific set of applications of interest and only seek to confirm those in decreasing order of weighted file %. Regarding the first use case, our M57 results indicate that present or previously present applications almost always score higher than non-present or never installed applications. The second use case also addresses part of the scalability question, in that our approach need not catalog a great number of applications in order to be of use, but rather only catalog applications of interest to the analyst.

5. CONCLUSIONS AND FUTURE WORK

In this work, we leveraged an existing catalog of full-file artifacts from specific applications to detect and reason over matching sectors recovered from media of interest. We used these matching sectors to suggest past uninstalled applications on test images and the drive images of the M57 Patents Scenario. Our results suggest that:

- Partial file contents (traces) remain after files are deleted due to application uninstallation
- These traces can be used to suggest the past presence of uninstalled applications

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Current approaches to determine prior application presence rely on intact artifact recovery, log analysis, Windows registry analysis, and trace evidence analysis. Our approach complements these methods, especially when intact artifacts and traces are not available and the Windows registry has been cleaned or is unavailable, e.g., on a non-Windows system.

Our approach requires that applications of interest be processed into the catalog prior to trace detection and computation. While processing new applications is relatively straightforward, it does require resources as well as knowledge of, and access to, applications of interest. Additionally, utilities that overwrite unallocated space would likely defeat our approach as we rely on fragments of deleted files residing in this unallocated space. Our approach is also vulnerable to deliberate deception, as the placement of specific file fragments in the unallocated space of a device or image, or even the creation and deletion of selected full-file artifacts, would cause spurious suggestion of an application that in fact had never been installed.

We are considering combining the weighted sector and weighted file measures, and also adding sector entropy and relative partial artifact location on the media to our measure of application presence calculation. Additionally, we are examining methods for more robust and precise noise reduction at the point of catalog creation, and we are considering sector differencing as an alternative to file differencing. Future work will extend our approach to malware applications and mobile platforms.

REFERENCES

Collange, S., Dandass, Y. S., Daumas, M., & Defour, D. (2009). Using graphics processors for parallelizing hash-based data carving. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on* (pp. 1-10). IEEE.

Dandass, Y. S., Necaise, N. J., & Thomas, S. R. (2008). An empirical analysis of disk sector hashes for data carving. *Journal of Digital Forensic Practice*, *2*(2), 95-104.

NPS-DEEP. (2015). Hashdb. Last accessed 10.4.15, https://github.com/NPS-DEEP/hashdb.

Forte, D. V. (2004). The "Art" of log correlation: Tools and Techniques for Correlating Events and Log Files. Computer Fraud & Security, 2004(8), 15-17.

Foster, K. (2012). *Using distinct sectors in media sampling and full media analysis to detect presence of documents from a corpus* (Doctoral dissertation, Monterey, California. Naval Postgraduate School).

Garfinkel, S. (2012). Digital forensics XML and the DFXML toolset. *Digital Investigation*, *8*(3), 161-174.

Garfinkel, S. L. (2013). Digital media triage with bulk data analysis and bulk_extractor. Computers & Security, 32, 56-72.

Garfinkel, S. L., & McCarrin, M. (2015). Hash-based carving: Searching media for complete files and file fragments with sector hashing and hashdb. *Digital Investigation*, *14*, S95-S105.

Garfinkel, S., Nelson, A., White, D., & Roussev, V. (2010). Using purpose-built functions and block hashes to enable small block and sub-file forensics. *digital investigation*, *7*, S13-S23.

Garfinkel, S., Nelson, A. J., & Young, J. (2012). A general strategy for differential forensic analysis. *Digital Investigation*, *9*, S50-S59.

Koppen, J., Gent, G., Bryan, K., DiPippo, L., Kramer, J., Moreland, M., & Fay-Wolfe, V. (2013). Identifying Remnants of Evidence in the Cloud. In *Digital Forensics and Cyber Crime* (pp. 42-57). Springer Berlin Heidelberg.

Laamanen, M., Nelson, A. (2014). NSRL Next Generation - Diskprinting. Forensics @ NIST, Gaithersburg, MD, December 3, 2014. Last accessed 10.4.15, http://www.nsrl.nist.gov/Documents/Diskprints.pdf.

Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). The art of memory forensics: detecting malware and threats in windows, linux, and Mac memory. John Wiley & Sons.

Nelson, A., Laamanen, M., Tebbutt, J., Long, D. (2014) Indexing the Windows® Registry for Software Detection. The American Academy of Forensic Sciences 66th Annual Scientific Meeting , February 20, 2014, Seattle, WA. Last accessed 10.4.15, http://www.nsrl.nist.gov/Documents/20140220%20Diskprint%20AAFS.pdf.

Nelson, A. J., Steggall, E. Q., & Long, D. D. (2014). Cooperative mode: Comparative storage metadata verification applied to the Xbox 360. *Digital Investigation*, *11*, S46-S56.

NIST. (2015). Diskprint Data Downloads. Last accessed 10.4.15, http://www.nsrl.nist.gov/dskprt/sequence.html.

NIST. (2012). Recommendation for Applications Using Approved Hash Algorithms, Special Publication 800-107 Revision 1. 2012. Last accessed 10.5.15. http://csrc.nist.gov/publications/nistpubs/800-107-rev1/sp800-107-rev1.pdf

Quick, D., & Choo, K. K. R. (2013). Digital droplets: Microsoft SkyDrive forensic data remnants. *Future Generation Computer Systems*, *29*(6), 1378-1394.

Roussev, V., & Quates, C. (2012). Content triage with similarity digests: the M57 case study. *Digital Investigation*, *9*, S60-S68.

Wong, L. W. (2007). Forensic analysis of the Windows registry. *Forensic Focus*, *1*.

Woods, K., Lee, C. A., Garfinkel, S., Dittrich, D., Russel, A., & Kearton, K. (2011). Creating realistic corpora for forensic and security education. ADFSL Conference on Digital Forensics, Security and Law.

Young, J., Foster, K., Garfinkel, S., & Fairbanks, K. (2012). Distinct sector hashes for target file detection. *Computer*, (12), 28-35.

Zobel, J., & Moffat, A. (1998). Exploring the similarity space. In *ACM SIGIR Forum* (Vol. 32, No. 1, pp. 18-34). ACM.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Do We Trust Image Measurements?

Variability, Accuracy and Traceability of Image Features

Mylene Simon, Joe Chalfoun, Mary Brady, and Peter Bajcsy
Information Technology Laboratory
National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD 20899
{mylene.simon, joe.chalfoun, mary.brady, peter.bajcsy}@nist.gov

***Abstract*—The paper addresses the problem of understanding quality of image measurements extracted using widely used software libraries from large images. Image measurements (features) are extracted using software packages that vary in terms of programming languages, theoretical formulas for the same image feature, algorithmic implementations, input parameters, units of measurements, and definitions of image regions of interest. Our motivation is to quantify numerical variability of image features across software packages and determine image accuracy with respect to reference images. In addition, our objective is to enable scientists to extract any image feature of interest from heterogeneous software libraries and gain provenance of every extracted numerical feature value.**

We pursue this objective by designing a client-server system that integrates image feature extractions from open source libraries such as ImageJ/Fiji, Python (scikit-image), CellProfiler, WND-CHARM, and in-house Java software packages. The system becomes useful for evaluating quality of image measurements, leveraging distributed computational resources for feature computations over big image data, sharing resulting feature values, and reproducing the feature values based on provenance. As an application of the designed system, we report the quality evaluations of 319 image features extracted using ImageJ/Fiji, Python (scikit-image), CellProfiler and in-house Java software packages with 43 duplicate features across the four packages. Using the normalized difference as metric, we identified 6 out of the 43 common features to differ over 1% in value and discuss the sources of these numerical differences.

Keywords-Big Data Science and Foundations: Data and Information Quality for Big Data

I. INTRODUCTION

Quantitative imaging and image informatics depend on taking image measurements over a region of interest. These image measurements are frequently denoted as image or object features. They are extracted by applying a wide variety of mathematical operations to image pixel values implemented in software. The software implementations vary in terms of programming languages, theoretical formulas for the same image feature, algorithmic implementations, input parameters, units of measurements, and definitions of image regions of interest. The motivation of our work is (1) to understand the variability of image feature extractions and (2) to enable scientists to compute any feature of interest from widely-used heterogeneous software libraries and gain provenance of every extracted numerical feature value. The word provenance refers to the record of image feature value's ultimate computational derivation and passage through various software tools.

To study the variability, the challenges lie in (a) identifying the same image features in multiple software packages, (b) setting their input parameters consistently, and (c) establishing pairs of reference images and image features for evaluations. In addition to the knowledge about variability, one would like to gather and access information about each feature extraction so that the feature values are traceable, reproducible, and executable in parallel on big image data. The challenges of enabling extraction of traceable image features lie in (a) integrating heterogeneous software, (b) gathering and hyperlinking all provenance information about feature extraction, and (c) designing a client-server system that enables data upload, configuration, computationally scalable feature computation, and access to feature values and all provenance artifacts to re-execute the image measurements. All the above challenges are mapped into two basic questions and associated problems. (1) What image features are trustworthy across software packages? (2) What practical solution would improve our trust in image features?

We approach the first problem by considering 319 image features extracted using ImageJ/Fiji, Python (scikit-image), CellProfiler and in-house Java software packages with 43 duplicate features across the four packages. Using the normalized difference metric, we identified 6 out of 43 features to differ over 1 % in value. We analyzed the sources of these numerical differences for some features to raise the awareness of community users. We approach the second problem by designing a web system with (1) interfaces to loading images and extracting image features while utilizing distributed computational resources, (2) access to feature implementations in several software packages, (3) a provenance information gathering mechanism, and (4) feature values hyperlinked with all computational provenance artifacts.

While there is an abundance of image feature implementations, the quality of feature values in terms of accuracy, variation, and execution traceability has not been

Bajcsy, Peter; Brady, Mary; Chalfoun, Joe; Simon, Mylene.
"Do We Trust Image Measurements? Variability, Accuracy and Traceability of Image Features."
Paper presented at 2016 IEEE International Conference on Big Data, Washington, DC. December 5, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

evaluated. Many image features have been implemented in academic environments [1], [2], [3], commercial platforms [4], or in publicly available image libraries [5], [6], [7]. The use of these image features is primarily for classification (find the most discriminative or predictive features, given a certain number of classes) and for discovery (understand statistical and semantic image characteristics based on image features). In the context of discovery, our focus is on the quality of image feature with respect to a semantically meaningful object rather than low-level image descriptors (e.g., Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Histograms of Oriented Gradients (HOG), Local Phase Quantization (LPQ), Binarized Statistical Image Features (BSIF), Local Binary Pattern (LBP) or Local ternary patterns (LTP)). Thus, our analysis is about accuracy of those image features that describe intensity, shape, or texture properties of an object. Regarding computational scalability and traceability of image feature execution, we are leveraging work in the area of scientific workflows, such as the capabilities of Pegasus [8]. In comparison to many existing scientific workflows, our work integrates heterogeneous software using a common file interface as opposed to wrapping software to a pre-defined workflow programming interface. Furthermore, our work is filling the gap in traceability of image feature execution by designing a community resource for sharing and reproducing scientific measurements.

The novelties of the work are (a) in documenting image feature variability across four software packages and (b) in designing a software plug-and-play framework for adding image feature extraction plugins, conducting image feature comparisons, and for delivering image feature values hyperlinked with computational provenance information.

The paper is organized as follows. Sections II, III, and IV are devoted to variability, accuracy and traceability of image features respectively. Sections II and III consist of evaluation setup, numerical evaluation, and deeper analysis. Section IV focuses on design and capabilities needed to extract any image feature of interest from heterogeneous software libraries and gain provenance of every extracted numerical feature value. An overall summary is provided in Section V.

II. Variability of Image Features

The variability study provides numerical evidence about the differences in various implementations of the same image features.

A. *Feature Extraction Software*

We evaluated four software packages with the total of 218 unique features. The subsets of unique features are implemented in Python (40 features), ImageJ/Fiji (33 features), Java (74 features), and CellProfiler (101 features). Python features were implemented on top of an existing image processing library (scikit-image [5]), ImageJ/Fiji [6] features were implemented as a plugin using the ImageJ application programming interface (API), and Java features were implemented from scratch at NIST [9].

We focused primarily on intensity and shape features in this work.

B. *Test Images*

We chose the live phase contrast 3T3 images comprised of 238 images and a total of 8162 cells with different shapes and sizes (Figure 1) [10] to analyze common feature value variability between all 4 software packages. The cells are segmented using the EGT technique [11], and the masks are saved as labeled images. The features are computed on top of the segmented Regions of Interests (ROIs).

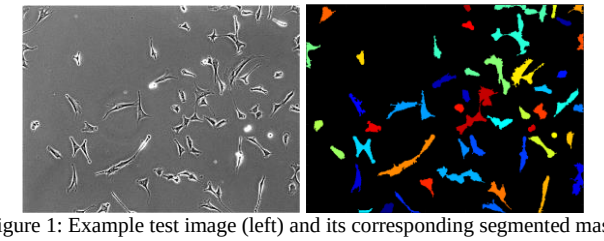


Figure 1: Example test image (left) and its corresponding segmented mask (right). Each ROI in the segmented mask has a unique randomly chosen color for display purposes.

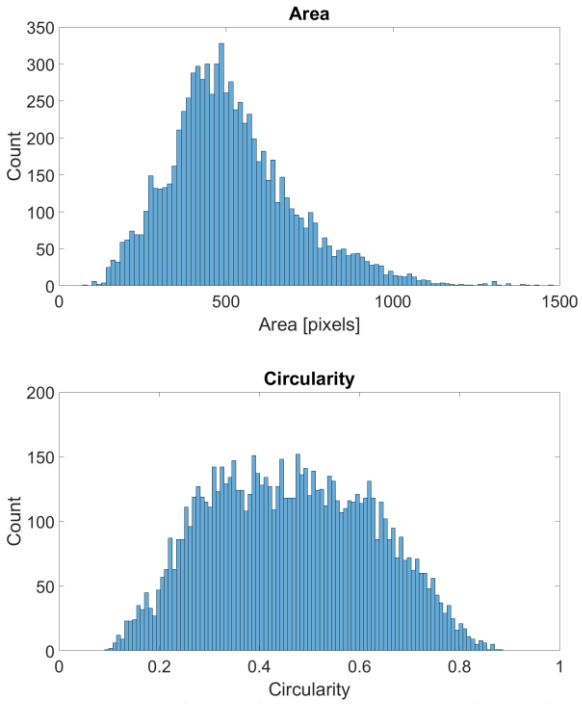


Figure 2: Histograms of area (left) and circularity (right) features from the objects defined by test images and their masks.

The measured images were selected for testing over synthetic images because (a) we did not have mathematical models for generating synthetic images that span a wide range of each image feature, and (b) we found the measured images to be a good initial approximation of the value range of each image feature. Figure 2 shows the histogram of test image measurements for area and circularity features.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

C. Evaluation Metric

Given two vectors of feature values V_1 and V_2 computed over a set of ROIs (image segments) by two software implementations of the same feature, we compute their dissimilarity metric S as the sum of relative errors $\boldsymbol{E_i^m}$ normalized with respect to the minimum of the two values from the vectors V_1 and V_2 that exceed a given threshold:

$$S = \sum_i (\boldsymbol{E_i^m} > T) \qquad (1)$$
$$\boldsymbol{E_i^m} = |\mathbf{V_{1i}} - \mathbf{V}_{2i}| / \min(\mathbf{V}_{1i}, \mathbf{V}_{2i})$$

The index i = 1,…,n and n is the number of ROIs. T is the user defined error threshold defined as 1 % of $\boldsymbol{E_i^m}$ in our work. The purpose of T is to detect substantial feature differences. The error is normalized by the minimum value which conveys the worst case error scenario.

D. Image Feature Variability Analysis

TABLE I. shows the results of feature variability evaluations using the aforementioned metric. The "Agree" column indicates when software have less than 1 % error across all 8162 test cells. The "Disagree" column indicates whether there is an error larger than 1 % across 8162 test cells between the tools that agree and the ones that disagree. The "Absent" column is used to denote with tools do not have an implementation of a given feature.

TABLE I. SUMMARY OF COMMON FEATURE VARIABILITY BETWEEN TOOLS BASED ON METRIC S (I = IMAGEJ, J = JAVA, P = PYTHON, C = CELLPROFILER).

Feature name	Agree	Disagree	Irrelevant
1- Perimeter		P,I,J,C	
2- Solidity	P, C	I	J
3- Circularity		I, J	C, P
4- Kurtosis		I, J	C, P
5- Bounding_Box_X	P, J	I	C
6- Bounding_Box_Y	P, J	I	C

Figure 3 illustrates the perimeter differences D_{ji} between its feature value V_{ji} and the average m_i of all three computed perimeter values per region of interest (i.e., cell segment). The feature difference follows the formula below:

$$D_{ji} = (V_{ji} - m_i) \qquad (2)$$
$$m_i = \frac{1}{3}\sum_{j=1}^{3} V_{ji}$$

The index i = 1,…,n; j = 1, 2, 3, j is the software index, and n is the number of ROIs, The perimeter values range between 44.4 and 542.5 pixels in the set of randomly chosen 64 ROIs from the 8162 available ones. The number 64 corresponds to approximately the number of ROI present in two figures. This random data sampling is done only for visualization purposes.

E. Sources of Feature Variations

We have summarized our analysis of some of the sources of feature variations for the features listed in TABLE I. The summary also includes features requiring "special attention" since they are prone to variations.

Perimeter and Circularity: The perimeter variability comes from the fact that algorithmic implementations differ in counting interior or exterior pixels, use 4 or 8 connectivity of pixels, and might interpolate between the boundary points. Circularity is inversely proportional to perimeter squared.

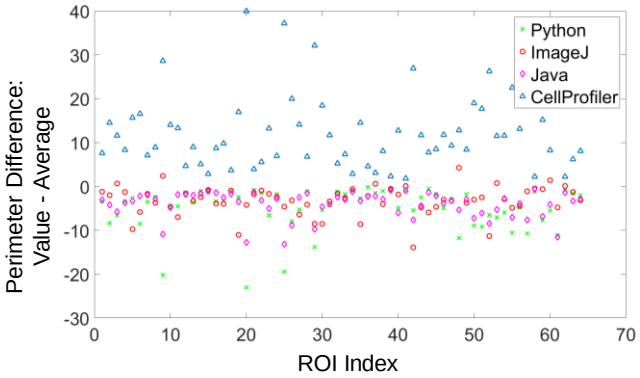


Figure 3: Perimeter feature differences over multiple regions of interests (ROIs). The unit is image pixel.

Solidity: The same definition of solidity is used by Python and ImageJ (Area/Convex Area). The difference between these values comes from the convex area differences since the implementations vary.

Kurtosis and Skewness: The kurtosis disagreement in values between software packages depends on whether the excess kurtosis or kurtosis are implemented (fixed offset by 3). Similarly, one has to be aware of multiple definitions of skewness, for instance, sample versus population skewness.

Centroid (and Bounding Box): The centroid and the bounding box are both subject to the choice of the reference coordinate system (+col ~ x; +row ~ y or +row~ -y). In addition, the bounding box of a ROI is defined by its upper left corner coordinate and its width and height. However, the bounding coordinates might be vary depending on the choice of values as integers or floats in a pixelated image.

Euler number (special attention): The Euler number definition is the number of objects (ROI) minus the number of holes. The value might differ depending on the assumptions about the number of ROIs (Python assumes #ROIs = 1).

Histogram bins for intensities represented by more than 8 bits per pixel (special attention): ImageJ uses the max

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

value plus one as the upper value of the last bin. It assumes that the lower value of the first bin is always zero.
Python and its numpy library provides two definitions. B = histogram(X, N) uses N equally spaced bins within the appropriate range for the given image data type. The returned image B has no more than N discrete levels. B = histogram(X,edges) sorts X into bins with the bin edges specified by the vector, edges. Each bin includes the left edge but does not include the right edge. The last bin is an exception since it includes both edges.

Orientation (special attention): The orientation is the angle between the major axis of a given ROI and the x-axis. It can be computed using two mathematical formulas: (1) $\theta = \text{atan}(\frac{V_y}{V_x})$ where atan is the arctangent function and V_x and V_y are the x and y decompositions of the major axis of the ROI; (2) $\theta = \frac{1}{2} atan2\left(\frac{2I_{xy}}{I_{xx}-I_{yy}}\right)$ where I_{xx} and I_{yy} are the second moment of area along the x and y axes and I_{xy} is the product moment of area. These two formulas are equivalent if the first one is computed in the range of $[-\pi/2, \pi/2]$ using atan and the second one in the range of $[-\pi, \pi]$ using atan2. The variations are observed if different value ranges or angular units would be reported by selected software packages. Range can be either $[-\pi/2, \pi/2]$ or $[-\pi, \pi]$ and the unit can be either radian or degree. In addition, the sign of the output angle depends on the coordinate system (image coordinate or graph coordinate system with clockwise or counter clockwise axes).

F. Discussion

Absolute value of feature variability depends on the choice of a metric. The current metric has a user-specified threshold that was set empirically to 1 % of the relative error E_i^m in our analysis. In addition, the current analyses do not include texture features because we encountered a large variety of definitions, naming inconsistencies, and hard-coded parameters. The work on including texture features is in progress. We would also note that the feature comparison is conducted using pixel units. Among the evaluated software packages, ImageJ reports all measurements in physical units (i.e., micrometers) that a user should be aware of.

III. ACCURACY OF IMAGE FEATURE IMPLEMENTATIONS

Accuracy analysis is based on two key components: (1) generation of synthetic images and their corresponding reference feature values, and (2) a metric to compute the error between reference and computed values. In this study, we used a theoretical feature value as the reference and assumed that the generated synthetic images are very close representations of analog shapes associated with the theoretical model. We documented the representation approximations by collecting and comparing image features for a range of analog shape parameters.

For the accuracy evaluations, we used the same software libraries as before but sub-selected image features for which we could generate reference feature values. Given the fact that we know the reference value, we could compute normalized relative errors E_i^r per ROI with respect to the reference feature value R_i (in comparison to the minimum value used in Eq. (1)).

$$E_i^r = |V_i - R_i|/R_i \quad (3)$$

V_i is the measured feature value, and i is the index of a ROI (image segment).

A. Test Images

We designed synthetic images with objects for which intensity, shape, and texture could be computed theoretically. Figure 4 shows three such image examples. The mathematical models include linearly increasing intensities with varying ranges and an ellipse shape with varying minor and major axes. The parameters of each model were varied when generating synthetic images. Additional synthetic images were created for testing Euler number as shown in Figure 7.

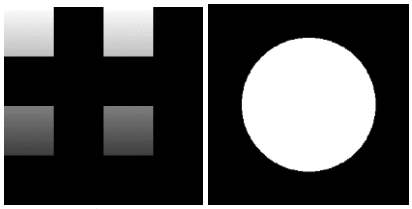


Figure 4: Examples of two synthetic images with known intensity and shape image features.

B. Image Feature Accuracy Analysis

We report accuracy analyses for three image features including (1) perimeter of a circle, (2) major and minor axes of ellipse, and (3) Euler number of a binary image.

Perimeter: We created 23 synthetic binary images of a circle with radius ranging between $r \in [2, 222]$ pixels. The circle generation is done on images with size (500, 500) pixels using the following formula:

$$if\ (x - x_c)^2 + (y - y_c)^2 \leq r^2\ then\ p(x, y) = 1 \quad (4)$$

The circle centroid coordinates are $x_c = 249$ and $y_c = 249$, and r is the circle radius. The perimeter reference value was set to $2\pi r$. Figure 5 displays the normalized perimeter error E_i with respect to the ground truth that is computed as a function of circle radius. Based on Figure 5, the error is large (up to 25%) for small radius values and it converges to a value of 5% as the radius reaches values larger than 100.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

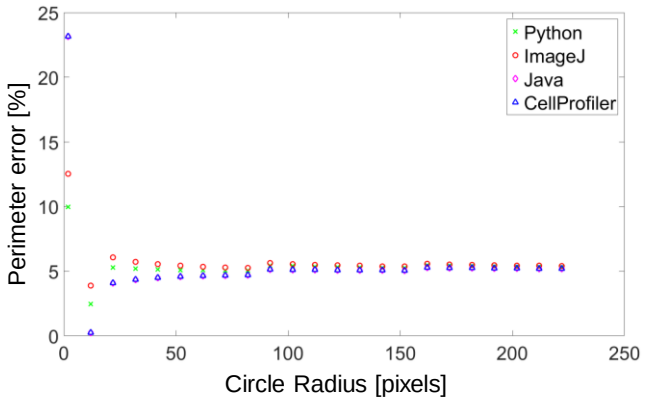


Figure 5. Normalized perimeter error vs circle radius

Major and minor axis length: During the feature variability evaluation, we detected minor differences that were below the 1 % threshold on feature error. To test major and minor axis lengths we created a set of 55 ellipse images with multiple values for major and minor axis length that ranges between 10 and 370. Figure 6 shows the normalized relative error E_i of major axis length computed according to Eq. (3) for ROIs in the 55 simulated images. It was observed that all three implementations had an error larger than 0.1 % when the ellipse shape was flat or the ellipse area was small. All implementations demonstrated the same dependency of feature error on ellipse shape/area.

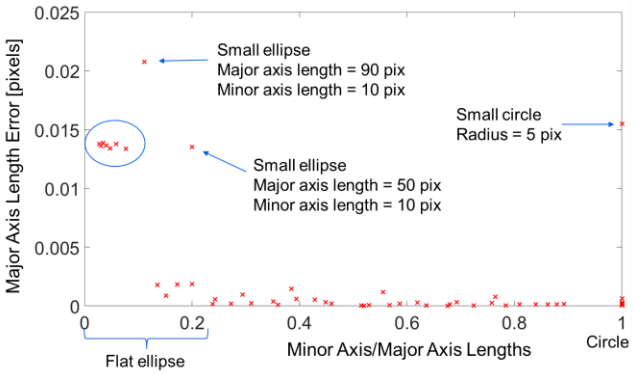


Figure 6. Major axis length normalized relative error as a function of minor/major axis length for 55 synthetic ellipses.

Euler number: Figure 7 displays three synthetic images and their Euler numbers (EN). EN is computed as the number of ROIs minus the number of holes. The values computed by Python deviated from the reference numbers since the implementation assumes only one ROI.

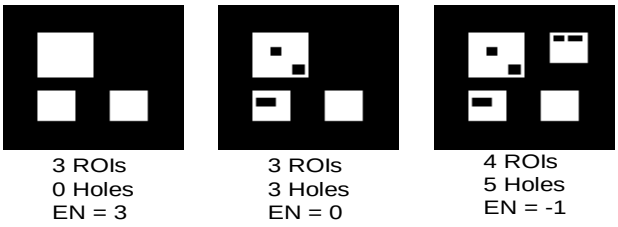


Figure 7. Synthetic images created for testing the Euler Number feature

C. *Discussion*

The design of a synthetic image generator plays a significant role in representing the theoretical value and is always limited by the integer image lattice. For example, the results in Figure 5 and Figure 6 would have been different if we placed the center of each ellipse at the lattice intersection as opposed in the middle of a square pixel (offset is 0.5). While the tools would still agree amongst each other, there will be bias between the computed major axis length value and the reference value. Thus, additional accuracy evaluations are required to determine error contributions of synthetic image generators.

IV. Traceability of Image Features

In the effort to provide access to traceable image features, we designed a client-server architecture of a web system shown in Figure 8. The main capabilities of this web-based framework for traceable image feature extraction are: (1) extensibility to include image feature extraction libraries written in any programming language via a file interface, (2) data management to upload collections and download feature values, (3) configuration and execution interface to image features registered in the system, and (4) collaborative access to traceable image feature values that are hyperlinked to their provenance information and all downloadable re-execution artifacts.

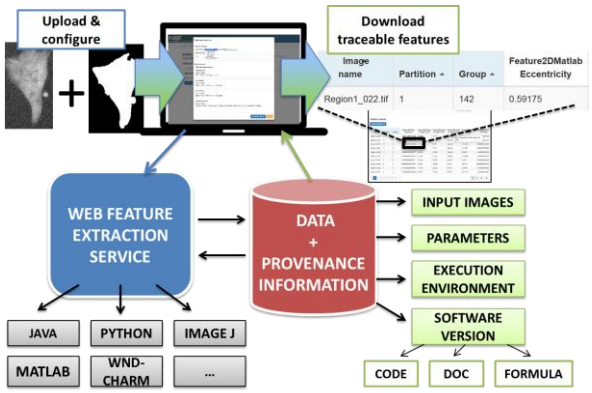


Figure 8: Architecture of a client-server system for traceable image feature extraction

The system was developed as a client-server application, built on top of a scientific workflow management system (WMS). The server-side follows the Java Representational State Transfer (REST) API built using the Spring

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

framework. It is coupled with a MongoDB database and calls Pegasus WMS [8] to manage the feature extraction jobs. The client-side is a light web application written in JavaScript using the AngularJS framework. The client side consumes the REST API from the server side.

A. *Integrating heterogeneous image software*

Ideally, one would like to load an image and its mask to RAM and then compute a spectrum of image features while capturing the provenance information. However, when dealing with image feature methods written in heterogeneous programming languages, the challenges lie in retrieving, compiling and executing feature methods on various platforms, and sharing image data loaded in RAM with each executable. Our approach is to regroup several existing image processing software packages into a single access point by designing a software integration framework. The other challenge of sharing data across heterogeneous software is addressed by using a common file interface to disk instead of more complicated sharing in RAM.

Input and output standardization: Our first step toward the homogenization of heterogeneous software packages was to define standard input and output formats.

Input: We chose the Extensible Markup Language (XML) to design a standard input interface defining the input parameters, such as the input file locations (raw images, optional segmentation masks, and optional tiling masks), output location, and feature extraction configuration (features to extract and their optional parameters values). Our choice of XML was motivated by the fact that most programming languages offer libraries that can be used for manipulating XML documents and the XML format is human-readable.

Output: We chose Comma-Separated Values (CSV) files as the standard output format for the extracted feature values. This format is both machine and human readable and supported in multiple software libraries. In order to merge feature values from diverse feature libraries, we defined naming conventions for the output file names and their CSV headers with input image, ROI and feature information.

Software executable and metadata integration: The system was designed to be extensible to any image feature implementation that can be run from a command-line. Each software is compiled to an executable that runs on a server and is launched with the common XML input file as an argument. In order to be integrated in the system, software has to be able to read the feature execution inputs from the XML input file and to write the feature results following the conventions we designed for the CSV output format and feature names.

The integration of a new *software executable* in the system is done by adding an entry in the Pegasus Workflow Transformation Catalog. The software is then registered in the MongoDB database by using the system API and saving its metadata in the database. The integration of new *software metadata* is performed by providing a list of features that the software is capable of extracting, along with eventual configurable options. The system can manage several versions of the same software, and each version corresponds to a new metadata entry in the database, linked to the corresponding executable via the Pegasus Transformation Catalog. The metadata stored in the database are used to generate the software documentation and feature selection web interfaces in the system, and to link the feature values to their provenance information.

B. *Key aspects of image feature execution*

In order to facilitate the execution of traceable image features implemented in heterogeneous software, we focused on (1) automated construction of the execution inputs captured by the XML input file, (2) input and output data management, (3) computational scalability of feature extraction, and (4) traceability of computed image feature values.

Construction of feature execution inputs. The input XML file is constructed automatically by collecting inputs via the client interfaces. Figure 9 shows one of the web interfaces to select features. The web interfaces allow a user to upload image and segmentation mask collections, select data and image feature implementations, and then launch the feature extraction. The selected features are linked to their software version metadata and the image collections involved in the extraction jobs are automatically locked on the server to prevent any further modification and allow the provenance information to be persistent.

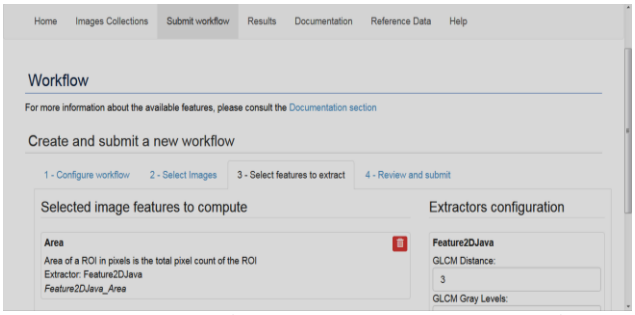


Figure 9: The web interface to selecting and extracting image features.

Data management: All input images, computed image features, job configurations and provenance information are stored in a MongoDB database. This database is accessed via the Java API on the server, allowing the registration of new software or new versions of existing software in the system and linking each computed feature value to all the provenance information.

Scalability of computations: The feature extraction computation is managed via the Pegasus scientific workflow [8], which was configured to use the High Throughput Computing workload management system HTCondor. Pegasus distributes the computations depending on the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

available computational resources, collects computational provenance, and monitors the execution.

Traceability of image feature values: The resulting image feature values from each software package are merged into a single table and delivered in a web interface. The delivered feature values are hyperlinked with the input images, the XML file that was constructed, the executable that was launched, the software repository with the version of the image feature extraction code, the execution environment provenance information, and the web documentation that contains the mathematical formula implemented in the code.

C. *Analysis of image feature traceability*

Image feature values are hyperlinked in the web interface with all artifacts in order to deliver feature traceability as defined in [12] (i.e., "ability to relate artifacts created during the development of a software system to describe the system from different perspectives.").

TABLE II. COMPLEXITY OF FEATURE RE-EXECUTION FROM ARTIFACTS FOR FOUR SOFTWARE PACKAGES. [JRE: JAVA RUNTIME ENVIRONMENT]

Image Feature Software	Complexity of Feature Re-execution	Required Additional Installation
In-house Java	Simple	JRE 1.7 or higher
ImageJ plugin	Medium	JRE 1.7 or higher and ImageJ or Fiji
Python program	Complex	Several libraries and packages with specific dependencies on a chosen operating system
CellProfiler wrapping program	Complex	Several libraries and packages, and CellProfiler v2.1.1 with specific dependencies on a chosen operating system

Beyond describing each feature value with artifacts, our ultimate test of feature traceability is to reproduce any feature value within a tolerance range from the hyperlinked artifacts [13]. This can be easily achieved by replicating the inputs (data and XML input file) and re-launching the feature computation in the same server environment. However, in order to run the downloaded executable with all inputs on a third party computer, one has to replicate the software environment of the server for launching the computation (i.e., install run time engines, scripting environments, libraries, and so on). We tested and classified feature re-execution from traceable artifacts outside of the server according to the complexity of additional installations as: simple, medium and complex. TABLE II. shows such evaluations of the three analyzed software packages.

D. *Analysis of Feature Computation Efficiency*

Having a client-server system for computing traceable image features provided us with a platform for comparing values across software libraries and against reference values as reported in Sections II and III. Here, we added a comparison of feature execution efficiency to complete the feature characterization.

We use 238 microscopy images of NIH 3T3 cells together with their manual segmentation masks from the set described in Section II.B. The raw images and masks are 16-bit per pixel images, each image has a file size of 446 kB and the entire data set contains a total of 8162 ROIs. In order to measure the performance of software packages against large images, we used one large field of view of stem cell colonies (22K x 22K pixels) imaged in phase contrast (16 BPP, 355.9 MB) and in Green Fluorescent Protein (32 BPP after calibration, 1.9 GB). The corresponding segmentation mask (16 BPP, 9.4 MB containing 200 ROIs) was obtained by segmenting the phase contrast image [14].

The following overlapping image features were calculated by all four software packages: Area, Perimeter, Mean intensity and Centroid. The calculations were run on a server with the specifications: Linux Virtual Machine (VM), operating Ubuntu 12.04 LTS 64-bit, four Intel Xeon CPUs and 16GB of RAM. The same calculations were run five times, and the timing and memory results were averaged.

The time and memory consumption breakdown per software is shown in TABLE III. as collected by the Pegasus workflow engine. CPU time is the time measured in system clock ticks for which the CPU was dedicated to feature calculations. This number is converted to seconds for our server with the number of clock ticks per second being 100 Hz. Max RSS (maximum Resident Set Size) is a measure of the memory occupied by a process and held in the main memory (RAM), given in kB (kilobytes) and converted in the table below in MB (megabytes)

Based on TABLE III. the in-house Java package has the fastest CPU computation time for a large number of small images (8162 ROIs in 238 images) with the speed-up factor of 2.81 (Python) and 36.55 (ImageJ). However, against a large image (GFP), Python is the most efficient, with a factor of 3.92 (ImageJ) and 2.51 (Java). We were not able to obtain results for CellProfiler due to its memory consumption exceeding our 10GB limit. Python is also the most efficient in terms of memory consumption, especially for the 3T3 dataset.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

TABLE III. AVERAGE CPU TIME, RUN TIME AND MAXIMUM MEMORY CONSUMPTION OF FOUR IMAGE FEATURE CALCULATIONS FOR 8399 ROIS ON 3T3 IMAGES (3T3), 200 ROIS ON PC IMAGE (PC), AND 200 ROIS ON GFP IMAGE (GFP) USING FOUR SOFTWARE PACKAGES.

Software	Dataset	CPU time [s]	Run time [s]	Max RSS [MB]
Python	3T3	48.82	58.00	49.98
	PC	35.71	36.38	3,731.41
	GFP	53.78	56.39	4,735.54
ImageJ	3T3	76.09	76.54	312.40
	PC	70.94	48.77	3,042.66
	GFP	211.12	183.66	5,070.66
Java	3T3	17.32	22.88	780.11
	PC	71.73	64.44	6,594.35
	GFP	135.38	186.02	7,292.10
CellProfiler	3T3	633.11	676.99	353.23
	PC	N/A	N/A	N/A
	GFP	N/A	N/A	N/A

V. SUMMARY

We quantified numerical variability of 43 overlapping image features across four software packages, determined image accuracy of 6 features with respect to their reference images and mathematical models, compared execution efficiency of 4 image features, and tested feature traceability in terms of complexity of artifact re-execution on third party hardware. These discoveries were enabled by designing a client-server system for integrating heterogeneous image feature libraries, executing feature calculations, and sharing hyperlinked image feature values with computational provenance artifacts. We deployed the system at NIST for internal use and testing.

In the future, we plan to complete the integration of additional publicly available feature extraction software in [1], [2] and [7] in order to make the image features traceable via our designed framework. We will also investigate the variability and accuracy of texture features, and disseminate the software and image feature studies to the image processing communities.

VI. ACKNOWLEDGMENT

We would like to acknowledge Antoine Vandecreme from the Computational Science in Metrology project at NIST who has contributed to the code development of the web image feature extraction system.

VII. DISCLAIMER

Commercial products are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

REFERENCES

[1] M. V Boland and R. F. Murphy, "A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells.," *Bioinformatics*, vol. 17, no. 12, pp. 1213–1223, 2001.

[2] N. Orlov, L. Shamir, T. Macura, J. Johnston, D. M. Eckley, and I. G. Goldberg, "WND-CHARM: Multi-purpose image classification using compound image transforms," *Pattern Recognit. Lett.*, vol. 29, no. 11, pp. 1684–1693, Aug. 2008.

[3] A. E. Carpenter, "Extracting rich information from images," *Methods Mol. Biol.*, vol. 486, pp. 193–211, 2009.

[4] "MATLAB and Image Processing Toolbox Release 2015b," *MathWorks Inc.* [Online]. Available: http://www.mathworks.com/help/images/index.html. [Accessed: 16-Mar-2016].

[5] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in Python.," *PeerJ*, vol. 2, p. e453, Jan. 2014.

[6] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nat. Methods*, vol. 9, no. 7, pp. 676–682, 2012.

[7] Itseez, "OpenCV (Open Source Computer Vision Library)," 2016. [Online]. Available: http://opencv.org/. [Accessed: 20-Mar-2016].

[8] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, "Pegasus, a workflow management system for science automation," *Futur. Gener. Comput. Syst.*, vol. 46, pp. 17–35, May 2015.

[9] I. NIST, "Image Features," *web page*, 2016. [Online]. Available: https://isg.nist.gov/deepzoomweb/stemcellfeatures. [Accessed: 31-Mar-2016].

[10] M. Halter, D. R. Sisan, J. Chalfoun, B. L. Stottrup, A. Cardone, A. A. Dima, A. Tona, A. L. Plant, and J. T. Elliott, "Cell cycle dependent TN-C promoter activity determined by live cell imaging," *Cytom. Part A*, vol. 79, no. 3, pp. 192–202, Mar. 2011.

[11] J. Chalfoun, M. Majurski, A. Peskin, C. Breen, and P. Bajcsy, "Empirical Gradient Threshold Technique for Automated Segmentation across Image Modalities and Cell Lines.," *J. Microsc.*, no. Under Review, pp. 1–18, 2014.

[12] G. Spanoudakis and A. Zisman, "Software traceability: a roadmap," in *Advances in software knowledge engineering*, vol. III, S. K. Chang, Ed. World Scientific Publishing, 2005, pp. 1–35.

[13] T. Crick, B. A. Hall, and S. Ishtiaq, "Reproducibility as a Technical Specification," *Comput. Res. Repos.*, vol. abs/1504.0, p. 6, 2015.

[14] K. Bhadriraju, M. Halter, J. Amelot, P. Bajcsy, J. Chalfoun, A. Vandecreme, B. S. Mallon, K. Park, S. Sista, J. T. Elliott, and A. L. Plant, "Large-scale time-lapse microscopy of Oct4 expression in human embryonic stem cell colonies," *Stem Cell Res.*, vol. 17, no. 1, pp. 122–129, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Measuring and Improving the Effectiveness of Defense-in-Depth Postures

Peter Mell
National Institute of Standards and Technology
100 Bureau Drive, Stop 8930
Gaithersburg, MD 20871
01-301-540-0061
peter.mell@nist.gov

James Shook
National Institute of Standards and Technology
100 Bureau Drive, Stop 8930
Gaithersburg, MD 20871
01-301-975-5264
james.shook@nist.gov

Richard Harang
Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783
01-301-394-2444
richard.e.harang.civ@mail.mil

ABSTRACT
Defense-in-depth is an important security architecture principle that has significant application to industrial control systems (ICS), cloud services, storehouses of sensitive data, and many other areas. We claim that an ideal defense-in-depth posture is 'deep', containing many layers of security, and 'narrow', the number of node independent attack paths is minimized. Unfortunately, accurately calculating both depth and width is difficult using standard graph algorithms because of a lack of independence between multiple vulnerability instances (i.e., if an attacker can penetrate a particular vulnerability on one host then they can likely penetrate the same vulnerability on another host). To address this, we represent known weaknesses and vulnerabilities as a type of colored attack graph. We measure depth and width through solving the shortest color path and minimum color cut problems. We prove both of these to be NP-Hard and thus for our solution we provide a suite of greedy heuristics. We then empirically apply our approach to large randomly generated networks as well as to ICS networks generated from a published ICS attack template. Lastly, we discuss how to use these results to help guide improvements to defense-in-depth postures.

Categories and Subject Descriptors
C.2.1 [**Computer-Communication Networks**]: General – *security and protection.*

General Terms
Algorithms, Measurement, Design, Experimentation, Security.

Keywords
attack graph, defense in depth, measurement, security.

1. INTRODUCTION
For network security, defense-in-depth (DID) is the security architectural practice of putting multiple defensive barriers between potential attackers and their desired targets [1]. These barriers often take the form of security devices or application portals/gateways; this then forms a layered security architecture. In this work, we measure the effectiveness of DID postures through evaluating residual weaknesses that, if exploited, can provide avenues for an attacker to breach the layers of security. We model the available avenues of attack both at the level of individual software vulnerabilities (e.g., Common Vulnerabilities and Exposures (CVE) names [2]) and at higher levels of abstraction (e.g., general threats that apply to classes of devices).

DID is a concept that is generally applicable in a wide variety of security domains. For example, DID has important application in industrial control systems (ICS) where computer controlled physical components are often separated from the Internet through a series of high level controllers and isolation devices [3]. It also has important application in cloud computing application architectures where, for example, a software-as-a-service (SaaS) application may have a series of layers that separate users from the raw application data [4]. It is also often important for storehouses of sensitive data where users must be limited in their access to the data (such as in federal government systems [5]).

We claim that an ideal DID architecture has three important features. It must be 'deep', meaning that there are many independent layers of security (we define independence later). It must be 'narrow', meaning that the number of node independent attack paths is minimized. In this work we focus on the equivalent but less intuitive idea of minimizing the smallest set of independent nodes whose removal could cut all attack paths. And it must be 'strong', meaning that each layer must provide the greatest possible deterrent to an attacker. In this work, we focus on measuring only depth and width because the strength of a layer is difficult to accurately assign. While strength is important to consider in one's layered defense, we focus here only on metrics that are quantitative and repeatable.

The basic definition of depth could be solved easily and quickly. However, the version of depth we need is computationally expensive to calculate. To evaluate it, we represent the DID security architecture as a type of attack graph. Each path in the graph represents a possible avenue of attack and each node represents a specific vulnerability or weakness on a host. The basic definition of depth asks for the shortest path from the initial node (point of expected hostile presence, often the Internet) to the closest target node. The target nodes vary by domain but represent the crown jewels of the network (e.g., the physical devices in an ICS or the main data store for a SaaS).

Complicating the basic definition of depth is that if an attacker can penetrate a particular vulnerability on one host then they can likely penetrate the same vulnerability on another host (given the same provisions for logical access). To account for this, we color each node corresponding to the vulnerability it represents and we then consider differently colored nodes to be independent. Thus,

Harang, Richard; Mell, Peter; Shook, James.
"Measuring and Improving the Effectiveness of Defense-in-Depth Postures."
Paper presented at 2nd Annual Industrial Control System Security Workshop (part of the 2016 Annual Computer Security Applications Conference), Los Angeles, CA. December 6, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

calculating depth is no longer a shortest path calculation on a graph (e.g., Dikjstra or Floyd's [6]), but instead a calculation of the fewest number of colors needed to enable a path from the source to one of the targets. We show that this small change to calculating shortest color paths (SCPs) converts the problem from a well-studied polynomial realm to being NP-Hard.

With respect to width, we use the same colored attack graph representation and calculate the smallest set of colors such that at least one color in the set will occur on each possible attack path. This is equivalent to finding the smallest set of colors where removal of all nodes of the chosen colors will break all attack paths. Without consideration for color, the attack graph could be transformed into a flow graph using published transformations and the minimum cut calculated in polynomial time [6]. However, we show that incorporating the colors and calculating minimum color cuts (MCCs) again converts the problem to being NP-Hard.

Since SCP and MCC are NP-Hard we developed a suite of approximation algorithms for them both. For some, we use variants on the standard polynomial time shortest path and minimum cut algorithms to attempt to greedily use many nodes of the same color (thereby minimizing the overall number of colors used). We also use genetic algorithms and another local minimum search approach. Lastly, we developed algorithms that provide exact answers, with limited scalability. These can be used on small problems and to test the accuracy of the approximation heuristics on such problems.

To empirically test our algorithms, we primarily use a set of randomly generated graphs to test effectiveness and scalability. We used another set of graphs to test applicability to an important DID domain area: ICS. For the effectiveness and scalability tests, we use layered colored attack graphs with 1000 nodes 10 layers. For the ICS domain testing, we generate ICS colored attack graphs by leveraging a published attack template of an ICS secured according to best security practices [7]. This attack template shows the security layers and attack paths which enable us to generate corresponding colored attack graphs.

From the random graph experiments, we compare the effectiveness and scalability of different heuristics. For the SCP, we find that our greedy breadth first search (BFS) algorithm provides the closest overall approximation to the actual SCPs. However, for less than 50 distinct vulnerabilities in the network, our exact algorithm can provide the true answer. For reduced runtime in this range, our genetic algorithm can be used and it empirically provides almost the exact answer. Our greedy BFS algorithm executes with linear time complexity and thus scales to massive networks. For the MCC, our greedy color-aware minimum node cut ensemble approach was the most effective overall. For greater accuracy with less than 100 vulnerabilities in the network, our genetic algorithm provides better approximations. For less than 30 vulnerabilities, our exact algorithm can provide the true answer.

From the ICS graphs generated from the ICS attack template, we performed a proof of concept of the approach on more realistic graphs. We determined that real ICS networks are layered (the attack template itself had 8 layers) and that our algorithms were able to determine the SCP and MCC which varied depending upon how sets of specific hosts were instantiated conformant with the attack template.

Our main conclusion is that he SCP and MCC metrics represent a novel way in which to measure characteristics of a DID posture. They are both quantitative and repeatable. While exponential to calculate exactly, we provide several scalable approximation approaches. While not intended to be a complete measure of security, SCP and MCC can contribute to an overall security metrics suite. In particular, they provide measurement of two particular features of the important principle of DID.

Our contributions in this paper include the following:

1. a novel and general method to express the DID of a network using layered colored attack graphs,
2. the novel metrics of SCP and MCC to measure DID effectiveness,
3. a proof that exact SCP and MCC measurements are NP-Hard,
4. and computational efficient and effective approximation algorithms for determining SCP and MCC.

The rest of the paper is organized as follows. Section 2 describes our colored attack graph representation for the DID architectures. Section 3 presents our security metrics of depth and width while section 4 describes our related algorithms. Section 5 discussed our test data: random layered security graphs and the ICS graphs. Section 6 provides the results of our empirical trials. Section 7 discusses how to use this data to optimize network security. Section 8 reviews related network security metrics and section 9 concludes.

2. REPRESENTING DID ARCHITECTURES AS COLORED ATTACK GRAPHS

To evaluate the DID posture of a network, we construct colored attack graphs that show available avenues of attack. These avenues can be general threats to specific types of devices, actual known vulnerabilities (e.g., CVEs), or a combination of both. In this section, we provide a novel attack graph construction using colored nodes that will enable us to evaluate DID characteristics.

Expected sources of attack are represented by a set of nodes, S, in our attack graph, G. A special 'super source' node, s, is added to G that has a directed edge to each node in S. The presence of s is for algorithmic convenience in reaching all expected sources of attack.

The high value targets in the network are represented by a set of nodes, T, in G. A special 'super sink' node, t, is added to G by adding a directed edge from each node in T to node t. As with s, t is added for algorithm convenience.

We then add a node to G for each vulnerability instance (or more generalized threat) in the network that could enable the creation of an attack path. Thus, each node represents a distinct vulnerability at a particular location (usually on some network host). For example, a node a may represent the vulnerability host pairing $[v_1,h_1]$ where v_1 is a vulnerability that exists on host h_1.

Edges are now added to G using the following formula (note, no additional edges are added to or from s and t). For each ordered pair of nodes, $a=[v_i,h_j]$ and $b=[v_k,h_l]$, a directed edge is created from a to b iff v_i on h_j provides an attacker sufficient privileges to exploit v_k on h_l and h_l is logically accessible from h_j. This accessibility will usually be network connectivity but may generalize to other forms of access (e.g., insertion of physical media in a device). For local attacks internal to a host, h_i and h_j may represent the same host.

Harang, Richard; Mell, Peter; Shook, James.
"Measuring and Improving the Effectiveness of Defense-in-Depth Postures."
Paper presented at 2nd Annual Industrial Control System Security Workshop (part of the 2016 Annual Computer Security Applications Conference),
Los Angeles, CA. December 6, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

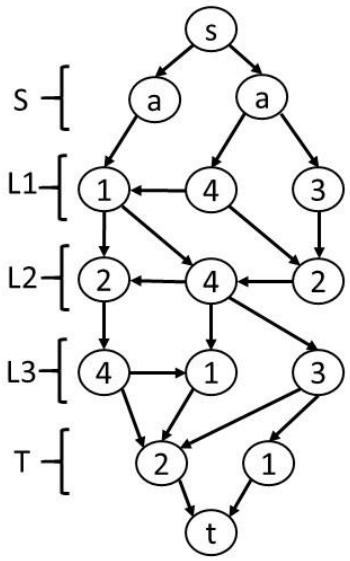


Figure 1. Example Colored Attack Graph Representing a Layered Security Architecture

An example attack graph is shown in Figure 1. At the top is the super-source and at the bottom is the super-sink, labelled with 's' and 't' respectively. On the left side one can see the groupings of nodes for S, T, and the individual security layers (denoted L1-L3) in the DID architecture. The layers are automatically generated by calculating the shortest path distance (e.g., Dikstra distance) minus 1 of each node from *s* and using that as the layer number. Thus, layer *i* contains all the nodes that are distance *i* from the set of attacking nodes.

The numbers within each node represent the color of the node. Note that the nodes in layer S do not have colors as they are the sources of attack and so they are labelled with an 'a'. The hostnames associated with each node are not shown. Note that multiple nodes may represent distinct vulnerabilities within the same host or between different hosts. Also note that actual layered security architectures can have edges traversing layers in both directions; we exclude such edges from our example for simplicity. However, edges may never traverse multiple layers as that would contradict our method of layer construction (such an edge would indicate a vulnerability that was put at the incorrect layer number based on its distance from *s*).

Our layered and colored attack graph representation where nodes represent host/vulnerability pairings is thus a novel, general, and flexible representation that can be applied to most DID architectures. One limitation is that our approach does not cover cases where an attacker must have privileges on two different hosts in order to execute an attack or where a combination of two or more exploited vulnerabilities is necessary to compromise another vulnerability. These are not the normal cases for vulnerabilities in public repositories (e.g., the National Vulnerability Database [8]).

While we have emphasized the novelty of our attack graph approach (with the colorings and layering), note that we leverage the node and edge semantics of the attack graph 'vulnerability oriented' model presented [9].

3. SECURITY METRICS

We can now use our colored attack graph representation to measure the DID posture of a network. We focus on measurements of depth and width. Depth measures the number of independent layers of security and width measures the smallest set of independent nodes whose removal could cut all attack paths. An increasing depth forces an attacker to penetrate more distinct vulnerabilities and a decreased width gives an attacker less flexibility in choosing which vulnerabilities to exploit. Thus, an ideal DID has both a large depth and small width.

Note that in this work, we do not combine these two measurements into a single DID strength score because such efforts tend to be ad hoc, even if operationally useful.

3.1 Depth – Evaluating Shortest Color Paths

To calculate depth, we must find a path from s to t that uses that fewest number of colors. We count only distinct colors instead of nodes because we assume that if an attacker can exploit a vulnerability on a host *a* then the attacker can exploit the same vulnerability on some host *b* (provided logical connectivity is available). The requirement for logical connectivity is easy to take into account because it is modelled by the edges in G. We refer to the entire operation of measuring the depth as finding the shortest color path (SCP).

Using the same example attack graph from Figure 1, Figure 2 shows the SCP by bolding, dotting, and making red the applicable directed edges.

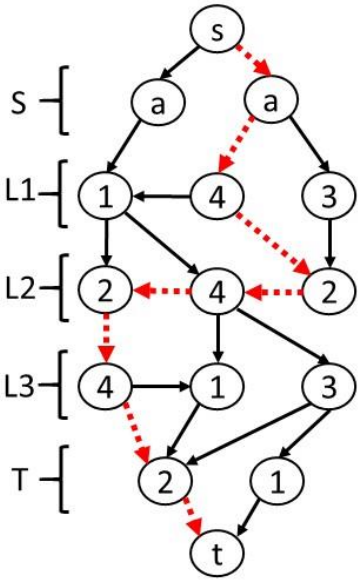


Figure 2. Example Attack Graph with the Shortest Color Path Highlighted

Note how the SCP is not necessarily the shortest path (as defined by number of nodes used). Here, the SCP is 2 (using colors 2 and 4) even though it traverses 8 nodes total. Remember that the nodes in S do not have colors since they are sources of attack. For comparison, the shortest path from s to t as traditionally defined is 6.

3.2 Width – Evaluating Minimum Color Cuts

To calculate width, we must find the smallest set of colors such that at least one color in the set will occur on every possible attack path. Another way to view this is to find a cut set of nodes with the fewest colors that eliminates all paths from s to t. We thus refer measuring the width as finding the minimum color cut (MCC). Note that the nodes in S are not candidates for the MCC as they represent the

attackers (they are not vulnerability options for the attackers to exploit). However, the nodes in T are eligible as they represent exploitation opportunities for the attackers on critical targets. Nodes s and t are not eligible as they were added for algorithmic convenience.

Using the same example attack graph from Figure 1, Figure 3 shows the MCC by bolding, dotting, and making red the applicable nodes.

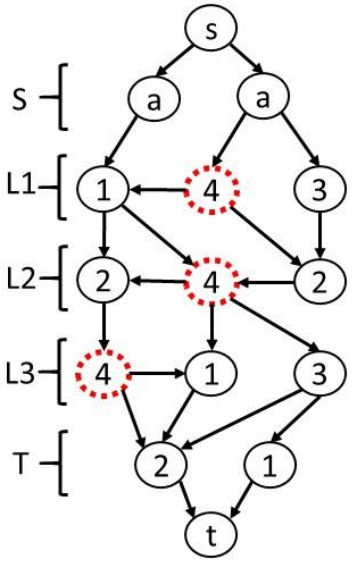


Figure 3. Example Attack Graph with the Minimum Color Cut Highlighted

Removal of the MCC nodes will disconnect s from t. Note how the MCC consists of all nodes of any chosen color (not just particular nodes that disconnect the graph). Thus, calculating the MCC is different from calculating the minimum cut (which counts the number of nodes used irrespective of color). The minimum cut for this graph is 2 while the MCC is 1 (even though it uses 3 nodes).

4. ALGORITHMS FOR DEPTH AND WIDTH

In this section, we present a variety of heuristics to calculate DID depth and width as well as high complexity exact approaches. Our best heuristic for SCP is a greedy shortest color paths algorithm. Our best heuristic for MCC is a color aware node cut algorithm. We also test a local minimization approach and a genetic algorithm for both SCP and MCC. The heuristics and genetic algorithms enable our measurements to scale to large networks while the exact approaches can be used for small networks.

4.1 Greedy Approaches

We used two greedy heuristics for our DID measurements. For depth, we use a greedy color-aware modified breath first search approach. For width, we make color aware use of standard minimum node cut algorithms.

4.1.1 Shortest Path Based Depth

To calculate depth we execute a breadth first search (BFS) algorithm [6] starting at s and terminating upon arrival at t. As typical with BFS algorithms, whenever a node is reached, it is labelled with its predecessor so that the shortest path can be enumerated upon reaching t by following the path in reverse. However, our BFS is modified to take into account color. Not only is each node labelled with its predecessor, but it is labelled with all colors used to get to that node.

Whenever a node is visited, the algorithm starts a secondary BFS from that node that is limited to visiting nodes colored with the previously used colors (those colors listed at the start node of the secondary BFS). This secondary BFS performs a greedy expansion but is not allowed to visit any nodes previously visited by any BFS instance (primary or secondary).

The secondary BFS capability enables a particular path being explored by the primary BFS to greedily include as many nodes as possible that use colors already used by the node currently being processed by the primary BFS.

Standard BFS is a linear time algorithm as is our variant (since each node will be processed by one and only one BFS instance) and so the computational complexity is $O(n+m)$ where n is the number of vertices and m is the number of edges.

4.1.2 Minimum Node Cut Based Width

To calculate width, we iteratively calculate the minimum node cut that separates the nodes in set S from t in the colored attack graph being evaluated. After each iteration, we remove all nodes of some chosen color. We repeat this until there does not exist a path from nodes in S to t.

There are several ways in which we choose the color to remove after each iteration:

1. Choose the color that is most frequent within the discovered node cut.
2. Restricted to the colors found in the standard node cut, choose the color that occurs most frequently in the entire attack graph.
3. Use method 1 above unless there is no color with an occurrence within the cut of greater than 1. In that case, use method 2.

Finding a minimum node cut (using the Edmonds Karp method) is $O(nm^2)$. We iteratively run this algorithm with the loop limited by the number of colors, c. Thus, our algorithms runs in $O(cnm^2)$.

We can assist the minimum node cut algorithm in identifying cuts with important colors by pre-processing the graph. Our pre-processor looks for nodes of the same color (e.g., a and b) that have edges to a common node (e.g., c) and adds a new node (e.g., d) with the color of a and b that is put in front of c. More specifically, the preprocessor creates a new node d, then adds edges a->d, b->d, and d->c, and then deletes edges a->c and b->c. This adds a single node that the node cut algorithm can find that allows both a and b to be cut from c. It makes the node cut algorithm somewhat color aware without modifying the algorithms itself. This preprocessor take $O(m+n)$ time.

4.1.3 Minimization Based Depth and Width

We also greedily calculate both depth and width through calculating which colors are necessary, taking us to a minimal solution. To do this, we rank the colors by their popularity (number of nodes with that color) and place them in a list. Then we iteratively remove the most popular color. We also create an, initially empty, set of 'necessary' colors which will be the output of the algorithm.

For each removed color, we construct a candidate color set of the remaining unprocessed colors combined with the necessary set. If the candidate set does not enable an (s,t) path (for SCP) or does enable an (s,t) paths (for MCC) then we know that the removed color is necessary and we add it to the necessary set. Processed

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

colors not added to the necessary set are dropped because they are not required for a color path (for SCP) or for a color cut (for MCC).

We can also execute this same algorithm for both SCP and MCC by providing an initial set of colors (colors not in the initial set are not used). This enables one to refine an existing SCP or MCC answer to find a minimal solution. Since the minimization routine executes quickly, we apply this optimization to the output of all our other approximation algorithms.

4.2 Genetic Algorithm Approaches

Our genetic algorithm (GA) approach represents all available colors as a bit string. It uses a fitness function that is the inverse of the sum of the number of bits set to 1 (i.e., number of colors used) multiplied by a binary value indicating whether or not there exists an s to t path using the allowed colors. We use a population size of 1000 and the initial population contains bit strings allowing all colors, all but one color, and bit strings where each color is turned on with a probability of .8. We thus bias our initial population towards using an excessive number of colors and allow the algorithms to trim out the unnecessary colors. To generate new populations, we leverage only bit strings from the previous generation that had a non-zero fitness function. We keep unchanged the top 10% scorers of the previous generation. The next 20% of the top scorers from the previous generation are kept with one of their colors randomly removed. Most of the remaining population for the new generation is created by using a random mask to crossover a random choice of the top 30% scoring bit strings with a random choice from all bit strings from the previous generation with a non-zero fitness. This process may generate candidate sets with duplicates (which we remove). We then ensure a new population size of 1000 by creating random bit strings (we always generate at least 10 random bit strings for each new population). To generate a set of random bit strings, we pick a threshold value for the probability of a bit being 1 (chosen uniformly from 0 to 1) and then use that value to generate all bit strings for that population.

Note that this algorithm takes a very pessimistic view of the usefulness of the graph since it only uses the graph to verify the correctness of proposed solutions. To verify a SCP solution, our linear time greedy approach is used. Because of this we can use the same algorithm to calculate both SCP and MCC with only a slight modification to the fitness function. For SCP, we force to 0 the fitness if an s to t path exists and for MCC we force it to 0 if an s to t path does not exist.

4.3 Exact Approaches

Since both SCP and MCC are minimization problems we can use a simple exhaustive search of the subsets of colors to find an exact solution. We use our previous approximation algorithms to determine an upper bound on the number of colors needed for SCP and MCC and then we iteratively try different sized sets of colors. At each chosen size, we exhaustively try all color combinations. We stop processing a particular sized color set if we find a set of colors that enable an (s,t) path (for SCP) or we find a set of colors that enable an (s,t) cut (for MCC).

We implemented three approaches. The first is to start with the upper bound and decrement the color set size to be searched by one in each iteration. The second is to use the upper bound to perform a binary search on the remaining possible color set sizes (reducing the possible set of sizes in half for each iteration). The third is to start at color set sizes of 1 and work upwards until we find the first working solution. The first approach empirically worked the fastest and so we use it exclusively in reporting results.

Note that these algorithms depend on the number of colors. If there are k colors, then the algorithms have to check up to 2^k subsets and then traverse up to n vertices for each subset. Thus, they are fixed parameter tractable problems. If the algorithms can be fed a good upper bound solution (e.g., from our approximation algorithms) then the exact algorithm can be sped up considerably which explains why our first approach outperformed the binary search approach in all of our experiments.

4.4 Complexity Evaluation

In this section we provide a proof sketch showing that both the SCP and MCC problems are NP-Hard (by using set cover reductions). Given that no polynomial algorithms exist to provide exact solutions for the set of NP-Hard problems, this explains our motivation to search for effective approximation algorithms.

We show that any instance of set cover problem can be reduced to an instance of an MCC problem in polynomial time. We let $U = \{u_1,\ldots,u_m\}$ and $S = \{S_1,\ldots,S_n\}$ be an arbitrary instance of a set cover problem where U is the set to be 'covered' and S is the set of sets that can be chosen to cover U. Assume that each S_i has cost 1. For an example problem for which we will show reductions, let U={a, b, c} and S={ac, ab, b}.

Given any set cover instance, we can create an auxiliary digraph. Let s and t be vertices. For each u_i, create a vertex disjoint (s,t)-path that has a vertex colored S_k for each S_k that contains u_i. Note that vertices s and t are not colored. This provides an auxiliary graph with m vertex disjoint (s,t)-paths. A minimum MCC calculation on the auxiliary graph will then yield a set of colors that correspond to the minimum set within S that covers U. Figure 4 shows the MCC auxiliary digraph for our example problem.

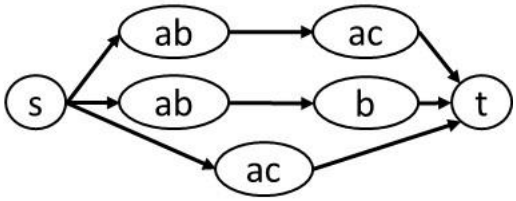


Figure 4. Auxiliary Digraph for Minimum Color Cut

The reduction of set cover to an SCP problem is as follows. Create nodes s and t. For each u_i, create a graph layer that contains a S_k colored node for each S_k that contains u_i. Including a layer for both s and t, this gives us m+2 layers (note that m=|U|) with each layer being an independent set. Create a complete set of edges from node s to the nodes in layer 1. For each layer $i< m$, create a complete set of edges from the nodes in layer i to layer i+1. Lastly, create a complete set of edges from the nodes in layer m to node t. The result is an auxiliary graph that is a layered graph as described in section 2. A minimum SCP calculation on the auxiliary graph will then yield a set of colors that correspond to the minimum set within S that covers U. Figure 5 shows the SCP auxiliary digraph for our example problem.

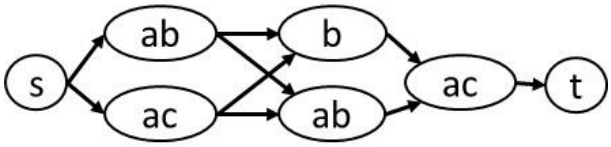


Figure 5. Auxiliary Digraph for Shortest Color Path

Thus, any arbitrary set cover problem can be reduced into both an SCP problem and an MCC problem such that a solution to either

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

problem yields a solution to the set cover problem. These are sufficient transformations to prove that the SCP and MCC problems are NP-Hard (the proofs themselves are not included due to space constraints).

5. DATA SETS

For our empirical analysis, we use two different data sets. The first is a random layered attack graph used to assess the effectiveness and scalability of our algorithms. The second is a layered attack graph based on an attack template for secured industrial control systems.

5.1 Random Layered Graphs

Given some enterprise network and starting from a set of possible attack sources (e.g., the Internet, employee desktops, or hosts that provide a SaaS interface), one can perform a breadth first search (BFS) of all possible attack paths from the attack sources to all other nodes in a network. In each iteration of the BFS, we consider the nodes reached to be at a new layer (representing increasing distance from the attack sources).

For our experiments we use this concept of layering to generate random attack graphs. We construct a graph by choosing the number of layers, the number of nodes per layer, the number of attack sources, and the number of high value targets. We add in the supersource 's' and supersink 't' as described in section 3. For edges, we decide on the probabilities for edges existing between layers of varying distances (including edges within a layer). The number of layers chosen will effect (but not decide) the depth of the graph and the number of nodes per layer will effect (but not decide) the width of the graph. By varying the number of layers and nodes per layer, we can scale the generated graphs.

5.2 Industrial Control System Data Set

An attack template for a hypothetical ICS that is secured with 'best practice' methods was provided by [7]. The template provides nodes for different actor types within the ICS (e.g., controller types, server types, and human entities). The edges representing possible attacks that can be launched from the vantage point of one actor in order to compromise another actor.

We use this attack template to generate randomized ICS attack graphs (colored and layered as described in section 3). The randomization comes in as we instantiate a particular actor type into actual instances of the actor. To perform this 'node explosion', we copy a single actor type node many times (retaining the same neighbors for each copy) in order to represent an instantiation of the actor type into actual actors. Each node copy is assigned a random color (unique from any color assigned to other actor types). We then take all node copies and connect them with edges to form a clique. Lastly we randomly remove edges incident with the copied nodes (both those edges used for the copied node clique and those edges with the rest of the attack graph).

This approach allows us to generate colored layered attack graphs conformant with the attack template but that randomly instantiate actor types into actors instances and randomly enable available attack edges from the template.

6. EMPIRICAL RESULTS

In this section we empirically evaluate the effectiveness of our proposed SCP and MCC algorithms as we scale the network in various dimensions. We use random layered attack graphs for the majority of the evaluation but then also apply our results to our instantiation of the ICS template.

We run our experiments in a virtualized Ubuntu operating system provisioned with a two Intel i7 cores and 10 GB of RAM. Our experimental system thus represents lower end commodity hardware.

6.1 Random Layered Graphs

We used random layered graphs to test the effectiveness and execution time as we scaled the number of colors, number of layers, and number of nodes per layer for both measuring SCP and MCC.

6.1.1 Shortest Color Path Measurements

We constructed random layered graphs with 10 layers, 100 layers per node, 100 attack sources (at layer 1), and 100 attack targets (at layer 10). The edge probability within a layer and between adjacent layers for each pair of nodes was 0.1. The attack sources in layer 1 represent a network where compromises are likely (e.g., where hosts communicate regularly with the Internet). The attack targets represent a layer of high value targets (e.g., hardware controllers in an ICS).

Using this model, we varied the size of the set of colors available. Each node was randomly assigned a color from the color set using a uniform distribution. This enables us to compare the relative effectiveness of our heuristics as the number of colors in a large graph increase.

We also ran experiments where we used the same setup but varied the number of layers and separately then number of nodes per layer (holding the number of colors constant at 50). There were no change in the relative effectiveness of the algorithms when varying the input along these dimensions.

Figure 6 shows the results for the Dijkstra 'naïve' approach, our SCP GA, and our SCP greedy BFS algorithm (our SCP algorithms with polynomial running times). Each data point represents the mean of 50 trials using different randomly generated graphs.

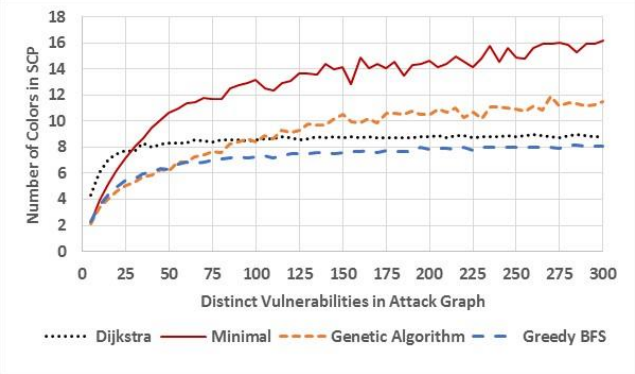


Figure 6. Relative Comparison of SCP Cardinality

The GA approach performs the best for up to 55 colors, after which the SCP greedy BFS provides the best solution. This is not unexpected because the GA is performing a multi-dimensional search on the bit string of color values. As more colors are added, the dimensionality of the problem increases making it more difficult for the GA to find effective local minimums. The naïve Dijkstra approach never provides the best solution although it converges toward the optimal as the number of colors increase. This is because once the number of colors equals the number of node in the network the problem converts into a standard shortest path problem. Likewise, the greedy BFS behaves more and more like the Dijkstra approach as the number of colors increases and thus also similarly converges to the optimal.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

We now move from comparing relative performance to a comparison against the exact SCP using the same experiment. Our algorithm to calculate the exact SCP has a combinatorial term which results in an exponential increase in execution time. Thus, Figure 7 shows the results only for small numbers of colors.

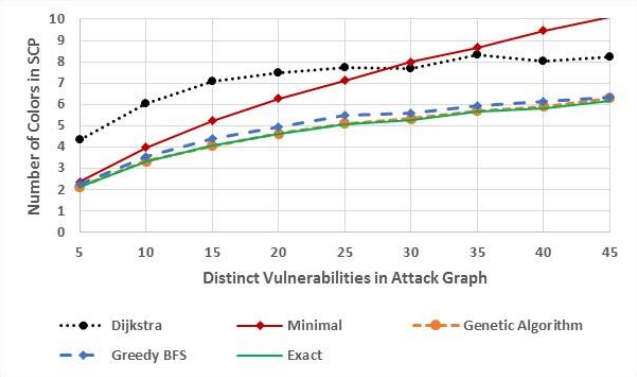


Figure 7. Exact Comparison SCP Cardinality

We see that the GA and greedy BFS approaches both provide close approximations to the exact SCP cardinality for the number of colors evaluated. The GA provided slightly better approximations in this color range. The naïve Dijkstra approach performed noticeably worse.

We next evaluate how the execution time for the algorithms changes as the number of colors increase, shown in Figure 8.

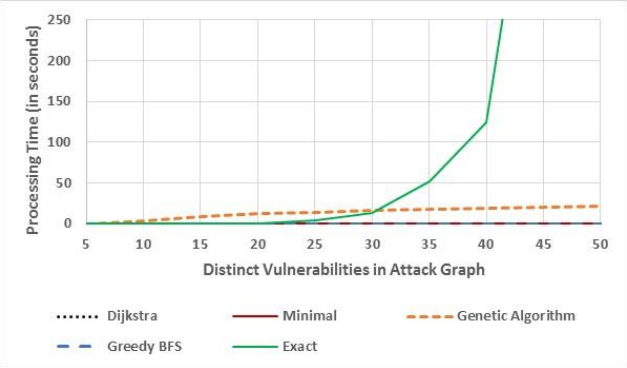


Figure 8. Growth of SCP Execution Time as the Total Number of Colors Increase

Here we see that the Dijkstra's algorithm and the greedy BFS operate essentially instantaneously. Both of them hug the x-axis and never take more than 0.01 seconds (even for 300 colors, not shown). The GA takes 41 seconds at 300 colors. The exact approach can clearly be seen to take exponential time making it tractable only for graphs with less than around 50 colors on our experimental setup.

6.1.2 Minimum Color Cut Measurements

We used the same experimental sets as with the SCP measurements and we apply it to compare the relative effectiveness of the MCC algorithms with an increasing number of colors.

As with the SCP measurements, we also ran experiments where we used the same setup but varied the number of layers and separately then number of nodes per layer (holding the number of colors constant at 50). There were no changes in the relative effectiveness of the algorithms when varying the input along these dimensions.

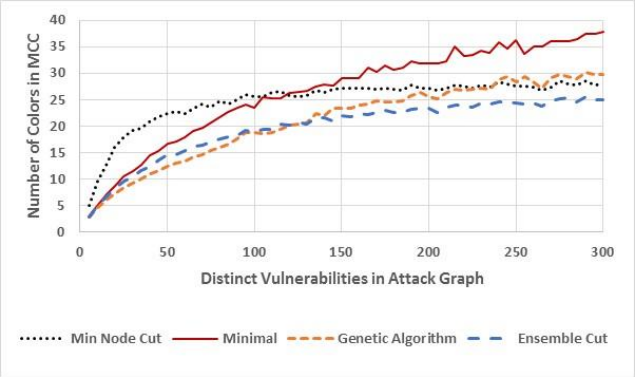


Figure 9. Relative Comparison of MCC Cardinality

As with the SCP GA, the MCC GA performs the best for a small number of colors but then loses performance as the dimensionality of the problem increases. For MCC measurements, this crossover happens at 130 colors. Our ensemble approach of color aware minimum node cuts performs the best after 130 colors but is reasonably close to the GA up to that point. The naïve minimum node cut approach always does much worse than our ensemble approach.

Next we compare the algorithms against exact answers, limited to a small number of colors since our algorithms to find the exact answers are exponential. Figure 10 provides the comparison up to 20 colors.

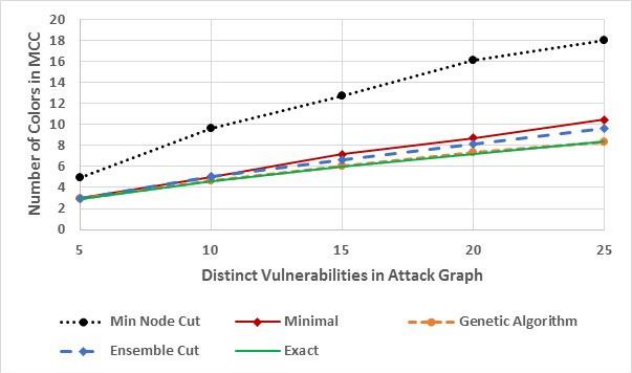


Figure 10. Exact Comparison of MCC Cardinality

We see that the naïve minimum node cut approach provides the worst performance. The GA so closely matches the exact answer that the lines are almost not distinguishable on the graph. The GA is at most 0.08 from the exact answer while. Our ensemble approach is never more than 1.25 away.

We next evaluate how the execution time for the algorithms changes as the number of colors increase, shown in Figure 11.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

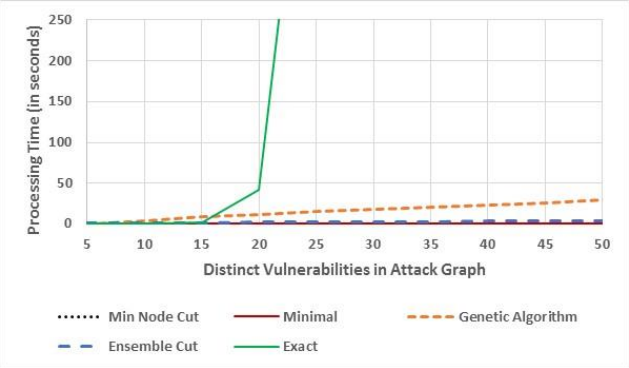


Figure 11. Growth of MCC Execution Time as the Number of Colors Increase

The exact solution clearly grows exponentially. The naïve minimum node cut approach takes at most 0.11 seconds and our color aware ensemble variant takes at most 6.8 seconds at 300 colors. The GA takes 80 seconds at 300 colors.

6.2 Industrial Control System Data Set

The ICS graphs were primarily used as a proof-of-concept test for both our SCP and MCC metrics as well as the algorithms used to implement them. These graphs were instantiated from an attack template representing an ICS secured to industry standards. Thus, each link in our instantiated attack graphs represented actual attack types that occur between the modeled services/entities.

From this exercise, we obtained evidence that real ICS networks are layered (the attack template itself had 8 layers) confirming our overall approach of processing attack graphs using a layered analysis.

Note that the number of layers in the attack template did not reveal the SCP of the instantiated graphs. Depending upon which of the instantiated entities has the vulnerabilities and which actually communicated, the SCP varied significantly. With fewer vulnerable nodes and less communication, the SCP tended to be longer while with a high presence of vulnerable nodes the SCP often was smaller than the number of layers in the attack template. This occurred because some of the attack segments crossed multiple layers.

The MCC of the instantiated graphs might at first appear to be equal to the MCC of the attack template. While a correlation exists, they are not necessarily equal. The MCCs of the instantiated graphs vary based on which of the attack paths from the template are actually available in the instantiated ICS graphs. Note the nodes must be both vulnerable and have the ability to communicate in order for them to be used in an attack path.

7. DISCUSSION

We envision our SCP and MCC metrics being used to provide an understanding of a network's DID posture with respect to expected sources of attack (be it external or internal). This will be most useful for networks with entry points that are threatened and that have 'crown jewel' servers to be protected.

A single network can be measured over time to determine the relative trending of the DID posture. The 'colors' returned from our algorithms can be used to highlight vulnerability types that, if fixed, could significantly improve the DID posture.

The SCP and MCC metrics can also be used to compare the DID posture between multiple networks of the same enterprise. This could be used for accountability purposes. It might be more impactful to use the comparison metrics to determine which networks need investment in patch/remediation technologies or in DID architectures.

8. RELATED NETWORK SECURITY MEASUREMENTS

A significant amount of work has been done examining the role of diversity in security. A large subfield of this research focuses on inducing diversity on the level of a single system [10] [11] [12]; this is beyond the scope of what we consider, however it could be considered as a method by which to generate additional 'colors' to introduce to our model.

On the network level, which is more directly applicable to our current problem, the work of [13] considers a simple Boolean measure of "survivability" that simply indicates whether an attacker possesses sufficient attacks to compromise all components of a particular network or not, without considering the possibility of multiple paths to a particular goal. A more sophisticated reduction of network security properties derived from diversity to coloring algorithms is explored by O'Donnell and Sethu [14]. They focus on the case of a fixed network topology, and consider the allocation of different version of software to different nodes within that network to impede the ability of an attacker to find a reliable attack path, in effect finding either a perfect coloring of the graph or a minimally defective coloring. They examine standard heuristics to obtain these goals, as well as the impact of adversarial activity during the coloring process on finding such colorings.

Within the context of ICSs, DID is typically presented as layering of independent security mechanisms, including 'soft' ones such as corporate policy and proper training. Depth of network topology, while occasionally mentioned [15], is rarely discussed in detail beyond simple approaches such as use of a "demilitarized zone" and firewalls between different segments of a network that includes ICS services [15] [16]. Our examination of heterogeneous devices as imposing additional burdens on an attacker has not yet been considered in this context, to the best of our knowledge.

Our work bears some relationship to traditional attack graph work, as does much other work in risk estimation and mitigation in ICS and SCADA systems [17]. Indeed, much work has been done on various minimization problems associated with attack graphs. In their study of properties of attack graphs, Jha et al. [18] consider a question related to our MCC property in examining the smallest subset of some set of defensive measures whose implementation makes the network under consideration secure; by reduction to minimum hitting set, they show that this property too is NP-hard to satisfy. Wang et al. [19], focus on the preconditions required to trigger the various exploits or vulnerabilities within a network, and examine ways to minimize the cost of removing these pre-requisites and thus denying access to attackers. While many results are analogous, their attack graphs differ significantly from our approach in two significant ways. First, they typically consider only known attacks, and do not attempt to model risk inherent in novel attacks, as we do by considering device heterogeneity (although see the work of Ingols et al. [20] which briefly considers services and programs that could be targeted by a "zero day" attack). Second, they generally do not directly examine questions of depth in any manner analogous to our SCP approach, in which the adversary must transit through multiple heterogeneous devices to reach their goal (although see [21], which does consider the impact of traffic filtering through multiple layers of a network as a component of defense in depth).

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

9. CONCLUSIONS

DID is an important security paradigm for many types of networks and is one component among many that is important for measuring overall security. We have identified depth, width, and strength as three important characteristics of DID security.

In this work, we provided a novel and general method to express the DID of a network in terms of security depth and width. We modeled a network's DID characteristics using a novel colored attack graph approach that exposed the residual risks in a layered security architecture. The colors represented vulnerability types and enabled us to model the ability of an attacker to exploit a particular vulnerability if they have already successfully exploited that same vulnerability elsewhere in the network.

To measure depth and width, we provided the novel metrics of SCP and MCC along with a proof sketch showing that exact SCP and MCC measurements are NP-Hard. We leave evaluations of strength (the third characteristic) for future work given the difficulty of measuring it rigorously and defensibly. Finally, we provided computationally efficient and effective approximation algorithms for determining SCP and MCC.

We empirically tested our approaches on large randomly generated layered security architectures to show scalability and functionality for large networks. We also tested against ICS networks using a published ICS attack template to randomly generate conformant layered security architectures. This demonstrated functionality and applicability to one important DID domain.

Overall, the most effectiveness algorithms were the greedy shortest path algorithm for SCP and our ensemble of color aware node cut algorithms for MCC.

Performing these measurements can enable one to compare the relative DID security of a network over time or compare two different networks. This analysis can also guide enhancements to further strengthen a DID posture and thus increase overall attack resistance in critical and important infrastructure.

10. ACKNOWLEDGMENTS

This research was sponsored by the U.S. Army Research Labs and the National Institute of Standards and Technology (NIST). It was partially accomplished under Army Contract Number W911QX-07-F-0023. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, NIST, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation hereon.

11. REFERENCES

[1] U.S. National Security Agency, "Defense in Depth," [Online]. Available: https://www.nsa.gov/ia/_files/support/defenseindepth.pdf. [Accessed 25 06 2015].

[2] MITRE, "Common Vulnerabilities and Exposures," [Online]. Available: https://cve.mitre.org/.

[3] U.S. Department of Homeland Security, "Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies," 10 2009. [Online]. Available: https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/Defense_in_Depth_Oct09.pdf.

[4] B. Lyons, "Applying a Holistic Defense-in-Depth Approach to the Cloud," 25 07 2011. [Online]. Available: https://www.niksun.com/presentations/day1/NIKSUN_WWSMC_July25_BarryLyons.pdf.

[5] S. Jordan, "Defense in depth: Employing a layered approach for protecting federal government information systems," 16 11 2012. [Online]. Available: http://www.sans.org/reading-room/whitepapers/bestprac/defense-depth-employing-layered-approach-protecting-federal-government-information-system-34047.

[6] S. Even and G. Even, Graph Algorithms, Cambridge, 2011.

[7] E. Byres, A. Ginter and J. Langill, "How Stuxnet Spreads - A Study of Infection Paths in Best Practice Systems," 2011. [Online]. Available: http://www.controlglobal.com/assets/11WPpdf/110228_Tofino_Stuxnet.pdf.

[8] "National Vulnerability Database," National Institute of Standards and Technology, [Online]. Available: http://nvd.nist.gov.

[9] P. Mell, R. Harang, "Minimizing Attack Graph Data Structures," in *Tenth International Conference on Software Engineering Advances*, Barcelona, 2015.

[10] G. S. Kc, A. D. Keromytis and V. Prevelakis, "Countering code-injection attacks with instruction-set randomization," in *Proceedings of the 10th ACM conference on Computer and communications security*, 2003.

[11] R. C. Linger, "Systematic generation of stochastic diversity as an intrusion barrier in survivable systems software," in *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences*, 1999.

[12] B. Cox and D. Evans, "N-variant systems: a secretless framework for security through diversity," *Usenix Security*, 2006.

[13] Y. Zhang, H. Vin, L. Alvisi, W. Lee and S. K. Dao, "Heterogeneous networking: a new survivability paradigm," in *Proceedings of the 2001 workshop on New security paradigms*, 2001.

[14] A. J. O'Donnell and H. Sethu, "On achieving software diversity for improved network security using distributed coloring algorithms," in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004.

[15] K. Stouffer, J. Falco and K. Scarfone, "Guide to industrial control systems (ICS) security," NIST, 2011.

[16] D. Kuipers and M. Fabro, "Control systems cyber security: Defense in depth strategies," United States Department of Energy, 2006.

[17] P. A. Ralston, J. H. Graham and J. L. Hieb, "Cyber security risk assessment for SCADA and DCS networks," *ISA transactions*, vol. 46, no. 4, pp. 583-594, 2007.

[18] S. Jha, O. Sheyner and J. Wing., "Two formal analyses of attack graphs," in *IEEE Computer Security Foundations Workshop*, 2002.

[19] L. Wang, S. Noel and S. Jajodia, "Minimum-cost network hardening using attack graphs," *Computer Communications* , pp. 3812-3824, 2006.

[20] K. Ingols, M. Chu, R. Lippmann, S. Webster and S. Boyer, "Modeling modern network attacks and countermeasures using attack graphs," in *Annual Computer Security Applications Conference*, 2009.

[21] S. Jajodia, S. Noel, P. Kalapa, M. Albanese and J. Williams, "Cauldron mission-centric cyber situational awareness with defense in depth," in *Military Communications Conference*, 2011.

[22] P. Mell, K. Scarfone and S. Romanosky, "NIST IR 7435: The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems," National Institute of Standards and Technology, 2007.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

General Methods for Access Control Policy Verification

Vincent C. Hu, D. Richard Kuhn
National Institute of Standards and Technology
Gaithersburg, MD, USA
vhu, kuhn@nist.gov

***Abstract*— Access control systems are among the most critical of computer security components. Faulty policies, misconfigurations, or flaws in software implementations can result in serious vulnerabilities. To formally and precisely capture the security properties that access control should adhere to, access control models are usually written, bridging the gap in abstraction between policies and mechanisms. Identifying discrepancies between policy specifications and their intended function is crucial because correct implementation and enforcement of policies by applications is based on the premise that the policy specifications are correct. As a result, policy specifications represented by models must undergo rigorous verification and validation through systematic verification and testing to ensure that the policy specifications truly encapsulate the desires of the policy authors. Verifying the conformance of access control policies and models is a non-trivial and critical task, and one important aspect of such verification is to formally check the inconsistency and incompleteness of the model and safety requirements of the policy, because an access control model and its implementation do not necessarily explicitly express the policy, which can also be implicitly embedded by mixing with direct access constraints or other access control models.**

Keywords— Access Control, Authorization, Policy, Policy Verification, Policy Testing, Policy Tool, Model Checking.

I. INTRODUCTION

Access control (AC) systems control which users or processes have access to which resources in a system. They are among the most critical of computer security components. AC policies are specified to facilitate managing and maintaining AC systems, therefore faulty policies, misconfigurations, or flaws in software implementation can result in serious vulnerabilities. However, the correct implementations of AC policies by AC mechanisms are very challenging problems. It is common that a system's privacy and security are compromised due to the misconfiguration of AC policies instead of the failure of cryptographic primitives or protocols. This problem becomes increasingly severe as software systems become more and more complex, and are deployed to manage a large amount of sensitive information and resources that are organized into sophisticated structures. Therefore, identifying discrepancies between AC policy specifications and their intended function is crucial because correct implementation and enforcement of policies by applications is based on the premise that the policy specifications are correct.

AC models are usually written to bridge the rather wide gap in abstraction between AC policies and mechanisms to formally and precisely capture the safety requirements that AC systems should adhere to. As a result, policy specifications represented by models must undergo rigorous verification and validation through systematic verification and testing to ensure that the policy specifications truly encapsulate the desires of the policy authors. Verifying the conformance of AC policies and models is a non-trivial and critical task. One important aspect of such verification is to formally check the inconsistency and incompleteness of the model and policy safety requirements, because an AC model and its implementation do not necessarily explicitly express the policy, which can also be implicitly embedded by mixing with direct access constraints or other AC models.

In this document, we discuss general approaches for the verification for AC models and the testing of model implementations by first defining standardized structures of AC models. We then demonstrate the expressions of AC models and safety requirements in formal specifications of model checkers for the use of black box and white box model verifications that verify the integrity, coverage, and confinement of the specified safety requirements against models. In addition, an efficient way of generating test cases for the implementation from a model is discussed.

This document is divided into seven sections. Section I states the purpose, of this document. Section II introduces the general concept of AC policy and model. Section III explains the elements of AC safety and faults. The focus of this document is presented in Section IV, which introduces main concepts for AC model verification and testing. Section V provides some AC system implementation considerations. Section VI present some major related works. Section VII is the conclusion to the document.

II. GENERAL AC MODELS

An AC model is a formal presentation of an AC policy enforced by the mechanism and is useful for proving theoretical limitations of an AC system so that AC

Hu, Chung Tong; Kuhn, David.
"General Methods for Access Control Policy Verification."
Paper presented at IEEE 17th International Conference on Information Reuse and Integration (IEEE IRI2016),
Pittsburgh, PA. July 28, 2016 - July 30, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

mechanisms can be designed to adhere to the properties of the model. Users see an AC model as an unambiguous and precise expression of requirements. Vendors and system developers see AC models as design and implementation requirements. On one extreme, an AC model may be rigid in its implementation of a single policy. On the other extreme, an AC model will allow for the expression a wide variety of policies and policy classes. In general, all **nondiscretionary** AC polices can be modeled by static, dynamic and historical Finite State Machine (FSM) models from one of the following classes:

a) Static model

Static policies regulate the access permission by static system states or conditions such as rules, attributes, and system environments (times and locations for access). Popular AC policies with these types of properties include ABAC [1], MLS[2], and RBAC[2]. These types of policies can be specified by **asynchronous** or **direct specification** expressions of an FSM model. The transition relation of authorization states is directly specified as a propositional formula in terms of the current and next values of the state variables. Any current state/next state pair is in the transition relation if and only if it satisfies the formula, as demonstrated in Example 1:

```
VARIABLES
    access_state : boolean;  /* 1 as grant, 0 as deny*/
    ……..
  INITIAL
    access_state := 0;
  TRANS /* transit to next access state */
      next (access_state) :=
       ((constraint_1 & constraint_2 & …… constraint_n) |
       (constraint_a & constraint_b & …… constraint_m)
  ……..);
```

Example 1 – static AC model

The system state of access authorization is initialized as the ***deny*** state and moved to the ***grant*** state for any access request that complies with the constraints of the rule corresponding with each constraint predicate (i.e., *constraint_1.*& *constraint_n*) in a rule, and stay in the ***deny*** state otherwise.

b) Dynamic model

Dynamic policies may include temporal constraints that regulate access permissions by dynamic system states or conditions such as specified events or system counters or N-person AC policy. An AC model with these types of properties specifies that accesses are permitted only by a certain subject to a certain object with certain limitations (e.g., object *x* can be accessed only no more than *i* times simultaneously by user group *y*). For example, if a user's role is a cashier, he or she cannot be an accountant at the same time when handling a customer's checks. This type of policy can be specified with **asynchronous** or **direct specification** expressions of an FSM model, which uses a variable semaphore to express the dynamic properties of the authorization decision process. Another example of dynamic constraint states is enforcing a limited number of concurrent accesses to an object. The authorization process for a user thus has four states: ***idle***, ***entering***, ***critical***, and ***exiting***. A user is normally in the ***idle*** state. The user is moved to the ***entering*** state when the user wants to access the critical object. If the limited number of access times is not reached, the user is moved to the ***critical*** state, and the number of the current access is increased by 1. When the user finishes accessing the critical object, the user is moved to the ***exiting*** state, and the number of the current access is decreased by 1. Then the user is moved from the ***exiting*** state to the ***idle*** state. The authorization process can be modeled as the following asynchronous FSM specification; example 2:

```
VARIABLES
    count, access_limit : INTEGER;
    request_1 : process_request (count);
    request_2 : process_request (count);
    …….
    request_n: process_request (count);
    /*max number of user requests allowed by the system*/
    access_limit := k;  /*max number of concurrent access*/
    count := 0; act {rd, wrt}; object {obj};
    process_request  (access_limit) {
       VARIABLES
          permission : {start, grant, deny};
          state : {idle, entering, critical, exiting};
       INITIAL_STATE (permission) := start;
       INITIAL_STATE (state) := idle;
       NEXT_STATE (state) := CASE {
           state == idle : {idle, entering};
           state == entering & ! (count > access_limit):   critical;
           state == critical : {critical, exiting};
           state == exiting : idle;
           OTHERWISE: state};
      NEXT_STATE (count) := CASE {
           state == entering : count + 1;
           state == exiting : count -1;
           OTHERWISE: DO_NOTHING };
      NEXT_STATE (permission) := CASE {
          (state == entering) & (act == rd) & (object == obj):
grant;
           OTHERWISE: deny;
           }
    }
```

Example 2 – dynamic AC model

c) Historical model

Historical policies regulate access permissions by historical access states or recorded and predefined series of events. Representative AC policies for this type of AC policies including Chinese Wall [2] and Workflow AC [2] policies. This policy class can be best described by **synchronous** or **direct specification** expressions of an FSM model. For example, the following Example 3 synchronous FSM specification specifies a Chinese Wall AC policy where there are two Conflict of Interest groups *COI1*, *COI2* of objects:

```
VARIABLES
   access {grant, deny};
   act {rd, wrt};
   o_state {none, COI1, COI2};
   u_state {1, 2, 3};
    INITIAL_STATE(u_state) := 1;
    INITIAL_STATE(o_state) := none;
    NEXT_STATE(state) := CASE {
      u_state == 1 & act == rd & o_state == COI1: 2;
      u_state == 1 & act == rd & o_state == COI2: 3;
      u_state == 2 & act == rd & o_state == COI1: 2;
```

```
    u_state == 2 & act == rd & o_state == COI2: 2;
    u_state == 3 & act == rd & o_state == COI1: 3;
    u_state == 3 & act == rd & o_state == COI2: 3;
    OTHERWISE: 1; };
NEXT_STATE(access) := CASE {
    u_state == 2 & act == rd & o_state == COI1: grant;
    u_state == 3 & act == rd & o_state == COI2: grant;
  OTHERWISE: deny; };
  NEXT_STATE (act) := act;
  NEXT_STATE (o_state) := object;
```

Example 3 – historical AC model

Note that in practice, the same AC policies may be expressed by multiple different AC models or expressed by a single model in addition to extra constraint rules outside of the model.

III. AC SAFETY AND FAULTS

Safety is the fundamental property of an AC system, which ensure that the AC system will not result in the leakage/blockage of permissions to an unauthorized/authorized principal. Thus, an AC system is safe if no privilege can be escalated to unauthorized or unintended principals, but the correct privileges are always accessible to authorized principals. Safety is specified through the use of restricted AC models that can be proven in general for that model describing the safety requirements of any configuration [3].

Among all the safety features, **Separation of Duties** (SoD) [2] are more dynamic than others. SoD refers to the principle that no user should be given enough privileges to misuse the system on their own. For example, the person authorizing paychecks should not also be the one who can prepare them. SoDs can be enforced either statically (by defining conflicting roles, i.e., roles which cannot be assigned to the same user) or dynamically (by enforcing the control at access time). AC faults compromise the safety, at semantic level, AC faults are usually caused by erroneous or inefficient representation of AC properties or permission algorithms. At a syntactic level, AC faults are simply caused by implementation errors in AC mechanism such as coding errors, or misconfigurations of AC systems. In general, AC faults can be categorized into the following classes.

Privilege leakage

Privilege (i.e. action and resource pair) leakage refers to situations in which subject is able to access resources that are prohibited by the safety requirements. Such leakage may cause either the privilege escalation from one resource domain or class to prohibited ones such as leakage from lower to higher ranks of MLS policy, or privilege leak such as from one role to other prohibited ones of an RBAC policy. Privilege leakage can be caused by mistaken privilege assignment directly or careless privilege inheritance indirectly.

Privilege blocking

Opposite to privilege leaking, a privilege blocking fault blocks a legitimate access to rightful resources. Privilege blocking can also occur when the properties of AC policy cannot render a *grant* or *deny* decision, or there is no available logic in the AC policy algorithm for evaluating the access request. Privilege blocking can also be a result of the deadlock of access rules specification where: a rule has a dependency on other rule(s), which eventually depend back on the rule itself such that a subject's request will never reach a decision because of the cyclic referencing.

Cyclic inheritance

Cyclic inheritance fault refers to the problem of privileges inheritance from other subjects(groups), which also in a chain of inheritance relation inherit back to the subject(group)'s privilege. For example, subject *x* inherits privilege from subject *y*, which inherit privilege from subject *z*, which inherits privilege from subject *x*. Cyclic inheritance leads to undecidable or infinite access evaluation process.

Privilege conflict

Unlike regular programming logic that a later value assignment of a variable overwrites the previous assigned value of the same variable, the rules of an AC policy normally have no precedence consideration in permission evaluation. In other words, AC rules will not be overwritten by other rules unless specifically allowed to. Thus, privilege conflicts appear when the specifications of two or more access rules result in the conflicting decisions of permitting subjects access requests by either direct or indirect (inherit) access assignments. In addition, when multiple policies are evoked for permission, conflicting decisions between policies may occur.

Multi-policies considerations

In an enterprise environment, it may be required to have AC policies specified independently by different collaborative or networked systems in the enterprise. Thus, an inter-system access request may be evaluated by more than one policy that the requesting subject is governed under. Thus, AC policy autonomy should also be preserved for secure inter-system access. Maintaining the autonomy of all collaborative system is a key requirement of the policy for inter-operation. The principle of autonomy states that if an access is permitted by an individual system, it must also be permitted under secure inter-system access. The principle of security states that if an access is denied by an individual system, it must also be denied under secure inter-system access. In a collaborative system, violations of secure inter-system access can be caused by adding inter-system privilege inheritance relations, for example, Fig. 1 shows that privilege *k* inherits privilege *j* through legal inter-system privilege inheritance (because both has the same privilege level *j*), which is granted in network *x* but denied in network *y*. These types of violations can be detected by checking for cyclic inheritance, privilege leakage and SoD violation. Thus, both security and autonomy can be characterized as safety requirements of a multi-policies AC system, which should be preserved during collaborations. A meta-policy is a policy that is usually applied for reconciling policy autonomy difference or to handle priorities of access decisions rendered from more than one policy. Thus, in addition to autonomy requirements, AC safety requirement may include priority model within the meta policy [4].

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

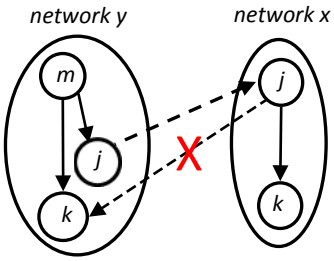


Fig.1 Privilege leaks through inter-system privilege inheritance.

IV. Verification Approaches

The fundamental goal of AC policy and implementation verification is to detect conflicting or missing rules (i.e. policy statements) by verifying AC policy model and testing output of the policy. To achieve this, semantically and syntactically methods with Black-box and/or White-box testing techniques may be used. Although the general safety computation is proven undecidable [5] especially for discretionary AC policies, which are impossible to be described by static policy models, practical safety constraints such as confinements can be specified for discretionary AC policies. As a result, verifications can be performed upon the constraints.

In a nutshell, AC policy verification must test if the **safety requirements** of an AC policy are incorporated in the expressed model, which will be the blue print for implementing the AC system. The specification of safety requirements can be AC properties, business requirements, specifications of expected/unexpected system security features, or direct translations of policy features. Safety requirements can also include privilege inheritance, for example to verify a SoD property, safety requirement will specify that 1) subject *x* and *y* are mutually exclusive if neither one inherit the other's privilege directly or indirectly, 2) If subject *x* and *y* are mutually exclusive, then there is no other subject inherits privilege from both of them. Similar to SoD, dynamic SoD (DSoD) has the safety requirement: 3) If SoD holds, then DSoD is maintained. Thus, 1) and 2) must be guaranteed [4].

Note that an AC policy is not necessarily explicitly expressed by a single model; it can also be implicitly embedded by mixing with direct access constraints or other AC models. Thus, an AC policy may be expressed by combining multiple AC models (e.g. for policy combinations) or additional constraints outside of the model into one combined model. The principle of ensuring the conformance of a model to the policy is to formally detect inconsistency and incompleteness faults as described in Section III. In the former case, for example, an access request can be both accepted and denied, while in the latter case the request is neither accepted nor denied according to the model.

Model Verification

The general approach for checking the correct specification of an AC model is to use black-box methods to verify the AC model against safety requirements. And since the confidence of the model's correctness depends on the quality of the safety requirements, a white-box property assessment method on entities in the model and safety requirements is required to assess the sufficiency of the safety, covering and confinement of the model [6].

In terms of AC attributes the formal definition of AC model can be illustrated by a deterministic finite state transducer of a model corresponding to a Finite State Machine (FSM) with a five-tuple $M = (\Sigma, ST, s_0, \delta, F)$, where Σ is the input alphabet that represents the attributes associated with subjects, actions, objects, and environment conditions.
ST is a finite, non-empty set of recorded AC system states and permissions, s_0 is the initial state, δ is the state-transition function, where $\delta : ST \times \Sigma \rightarrow ST$, F is the set of final states include *Grant*, *Deny* as the output.

For **static** AC models as described in II a, the FSM M_{static} does not require intern states to reach the permission state, thus $F = ST = \{Grant, Deny\}$, i.e., M_{static} is just a straightforward FSM model without state transitions. For **dynamic** AC models as described in II b, the input alphabets of FSM $M_{dynamic}$ are $\Sigma_{dynamic} = \{gCond_1, \ldots, gCond_n\}$, where global condition $gCond_i$ is the threshold indicator of the access limitation, such as the number of persons that have to access at the same time in a N-Person control policy [2], or the maximum number of accesses allowed for a Limited_Number_of_Access policy. For historical AC models as described in II c, the input alphabets of the FSM $M_{historical}$ are $\Sigma_{historical} = \{\ldots.. sCond_i, aCond_i, oCond_i \ldots\}$, where subject condition $sCond_i$ action condition $aCond_i$ and object condition $oCond_i$ contribute to a historical event that is used as determining factors for the next permission decision. Note that it is possible for different types of AC models to combine into one model such that $M_{combine} = \{M_{static} \cup M_{dynamic} \cup M_{historical}\}$[2].

An AC safety requirement p is expressed by the proposition p: $ST \times \Sigma^2 \rightarrow ST$ of FSM, which can be collectively translated in terms of logical formulae such that $p = (sCond_1 * \ldots * sCond_n * aCond_1 * \ldots * aCond_n * oCond_1 * \ldots * oCond_n * gCond_1 * \ldots * gCond_n) \rightarrow d$, where $p \in P$ and d is the permission is a set of safety requirements, and * is a Boolean operator in terms of logical formulas of temporal logic such as computational tree logic (CTL) and linear-time temporal logic (LTL). The purpose of model checking is to verify the set ST in M in which p is true according to an exhaustive state space search. In addition, by verifying the set of states in which the negation of p is true, we can obtain the set of counterexamples to make the assertion that p is true. The satisfaction of an AC model M to the AC safety requirement P by model checking is composed of two requirements:
(1) Safety, where M satisfies P. That is, there is no violation of rules to the logic specified in P, and it is assured that M will eventually be in a desired state after it takes actions in compliance with a user access request.
(2) Liveness, where M will not have unexpected complexities. That is, there is neither a deadlock in which the system waits

Hu, Chung Tong; Kuhn, David.
"General Methods for Access Control Policy Verification."
Paper presented at IEEE 17th International Conference on Information Reuse and Integration (IEEE IRI2016),
Pittsburgh, PA. July 28, 2016 - July 30, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

forever for system events, nor a **livelock** in which the model repeatedly executes the same operations forever.

Thus, the AC rules define the system behaviors that function as the transition relation δ in M. Then when the AC safety requirement is represented by temporal logic formula p, we can represent the assertion that model M satisfies p by $M \models Ap \rightarrow AXd$, where temporal logic quantifier A represents "always", and logic quantifier X represents "is true next state". The purpose of safety and liveness verification using model checking is to determine whether these assertions are true, and to identify a state in which the assertions are not true as a counterexample for the assertions. Since the behavior of the AC mechanism can be represented by FSM M, and the safety requirements that M must satisfy can be represented by temporal logic formulas, we can define the correctness more precisely as that the model can be led from every possible state that is reachable from initial states to the defined final state while complying with the safety requirements.

Even though checked by the black box testing as described above, the model is not fault proof because the temporal logic in the model might not be thorough in covering all possible values of all rules or all conditions in rules. For example, two states determined by opposite assignments of the same Boolean variable are embedded in different sub-state modules, where a third state is triggered only when the constraints of the two states are satisfied. As demonstrated in Fig. 2, the two rules will never agree due to the self-negation to the same constraint. In this case, the third state will never be satisfied, but proven correct without counterexamples through the black box checking.

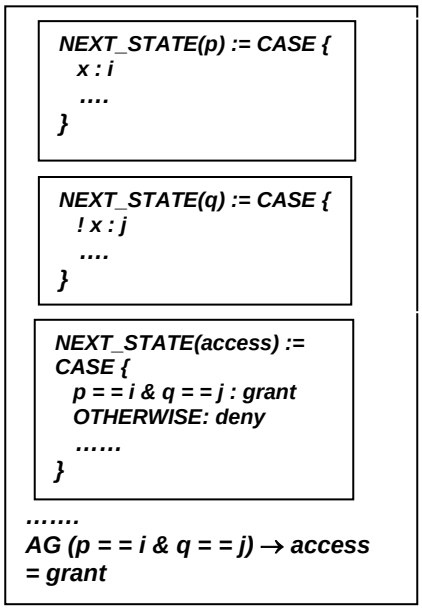


Fig.2 Example of never-achieved rules and the safety requirement in an AC model.

To detect this kind of semantic fault, the white box testing, based on code analysis should be applied such that the resulting mutated versions are used to detects faults of the model. Testing for mutations makes sure all paths of a part of a model code are covered by setting the related target variables to all possible values as input, and checking to see if there are different outcomes from the changes. If there is none, then either the code that had been mutated was never executed or the variable was unable to locate the faults. As shown in Fig. 2, If we mutate the first ***case*** module to change x to $!x$, the resulting *access* state will be *grant* without being affected. (That works the same for the second case module). This fault demonstrates that there is a redundancy in the model, which does not violate the temporal logic of the model. Further investigation to check the model that relates to the variable should reveal that the ***(p = = i & q = = j) → access = = grant*** safety requirement will never happen. Note that this fault can be caught if one more safety requirement ***E! (p = = i & q = = j)*** (which means there exists some path that eventually in the future will satisfy ***!(p = = i & q = = j)*** in CTL model checking) is specified for the black box checking. Hence, it is not expected that all safety requirements are perfectly specified in the beginning. Thus, white box checking can be used as a second line of defense against faults that will not be spotted by black box checking.

Most faults in a AC model result from the nondeterministic automata of FSM states, for example, in Fig. 3, white box checking will detect that the value x will result to a ***grant*** of ***access*** when it is either s or t. This does not violate the safety requirement, however, the safety property will not be maintained if a more stringent safety requirement requires that only one value of x attribute is desired from the policy.

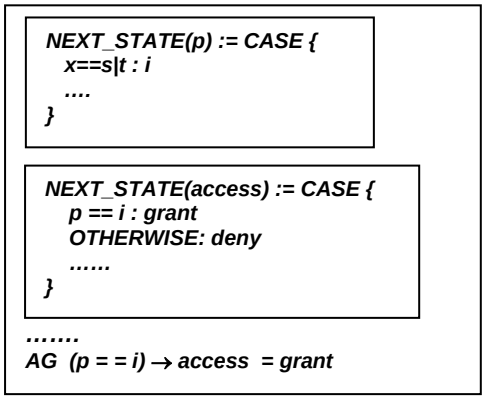


Fig. 3 Example of ambiguous value and the safety requirement in an AC model.

Another example shows a transition to an unspecified state for a certain range of data values such as in Fig. 4, there is no way for the black box checker to figure out the value of ***access*** when x value is other than s unless we check with the safety requirement ***AG ! (p = = i) → access = deny***. This uncovered value can be detected by the white box checking when different values were assigned to x, which does not match any expected case condition, and results in the same

grant of ***access***. Thus, the safety requirement verification informs the users which rules are not covered by the existing safety requirement so that the users can add new properties to cover the uncovered.

```
NEXT_STATE(p) := CASE {
   x==s : i
   ....
}

NEXT_STATE(access):= CASE {
   p == i : grant
   ......
}
.......
AG  (p == i) → access  = grant
```

Fig. 4 Example of uncovered value and the safety requirement in an AC model.

Coverage and Confinements Semantic faults

The rules in the policy, model, and safety requirements may each describe their own space of permission conditions, and may not be congruent in one space as the initial relation illustrated examples in Fig. 5. The safety and liveness check can assure only the logical integrity of some rules against some safety requirements. The complete satisfaction of a model to its policy requires fixing of coverage and confinement faults if any violations are detected by additional Coverage and Confinement Checks (CCC), the second line of defense against such semantic faults.

CCC requires mutant versions of the model, and extra modified properties for additional model checking. As illustrated in Fig. 5, the goal of CCC is to ensure that the rules in the safety requirement are completely covered by the model, and to confirm that no exceptional access permissions are granted unless intentionally allowed. The first step of CCC is to discover the rules, which are seeped through the specification of the safety requirement by applying white box checking on mutated versions of the model. The second step is to detect unexpected access permission that might not be the intention of the policy author, by applying model checking on modified rules extracted from the original ones.

Rule coverage checking

The key notion of rule coverage checking is to synthesize a version of the given model in such a way that the permission of its rules is mutated such that rule *r* is changed to ~*r*. If safety requirements are satisfied by both mutated and original models through model checking, then some of the rules and their mutants would never be applied to the safety requirements; in other words, the safety requirements do not cover all the rules in the model.

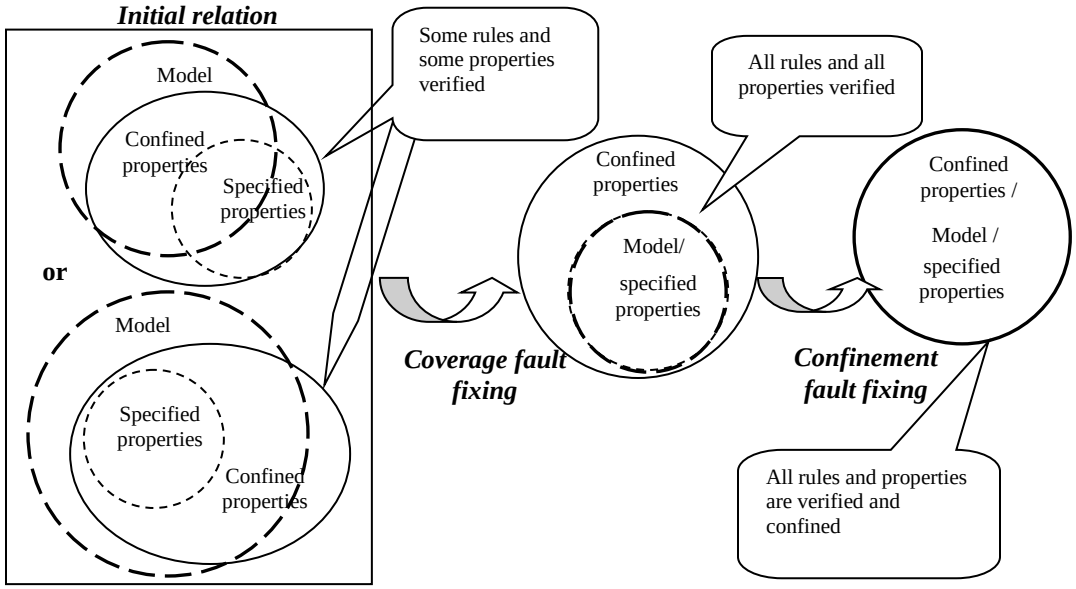


Fig. 5 Relations of Policy, Model, and Safety Requirements

As an example in Fig. 6, the safety and liveness checking verify that the model conforms the safety requirement ***AG (q = = i) → access = grant*** without counterexamples; however, by applying the CCC by mutating the rule ***u == j: grant*** to ***u == j : deny*** for the coverage checking, the result shows that the safety requirement satisfies the mutated rules as well (without counterexamples), indicating that the variable *u* was never applied to the safety requirement ***AG (q = = i) → access = grant***. This result shows that the rule ***u == j : grant*** is not verified with the property ***AG (q == i) → access = grant***. One way of addressing this insufficiency is adding a new property that describes proper control of *u*. Note that it is necessary to check every *rule in the model* against all safety requirement to achieve thorough verification.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

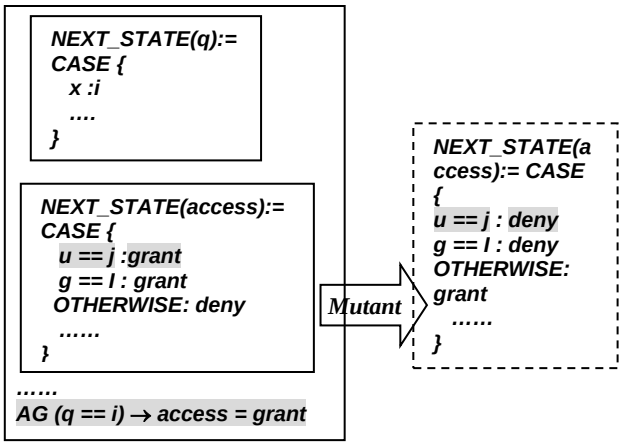


Fig. 6 Example of uncovered rules in a AC model

Property confinement checking

Property confinement checking ensures that there is no exceptional permission allowed in addition to the specified safety requirement; this checking requires a modified safety requirement to be added for the next run of model checking. Confinement check should discover the discrepancy of the specified safety requirement and the safety requirement the AC policy author intend. The rationale is that if the model does not satisfy the modified safety requirement, then there are exceptional access permissions that leak through the safety requirement. Fig. 7 shows a transition to an unspecified state for a certain range of data values that allow exceptional permissions not covered by a specified safety requirement because the value of ***access*** when *u* value is different than *i* (such as ***u = j***) also grants access permission by the rule ***otherwise: grant***. This fault can be caught by a counterexample ***AG (u == j) → access = grant*** when checking the model against the additional confinement property ***¬AG (u == i) → access = deny*** derived from original property ***AG (u == i) → access = grant.*** The additional model checking for confinement verification informs the AC policy authors which safety requirement is not confined so that the AC policy author can add new rules to enforce the safety of the model. As in this case, changing the rule ***otherwise: grant*** to ***otherwise : deny*** and adding all granted rules in the state will correct the problem.

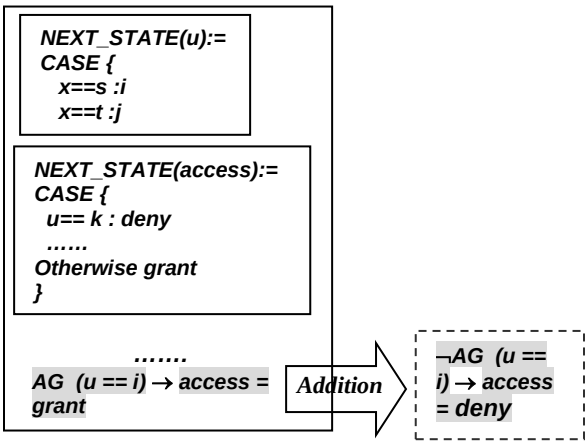


Fig. 7 Unconfined rule in a property

Note that it is possible the AC policy author intentionally allowed the exception for a safety requirement, and it is necessary to check every safety requirement against the set of rules in the model to achieve thorough verification.

Implementation Test

Black box model checking and white box mutation test provide methods for verifying the correct model representation of the policy. Once a model is verified, the AC mechanism can be implemented based on the design of the model and additional constraints if needed. Usually AC mechanisms are code developed in a language the AC system supports, for example dedicated AC language such as XACML [7] is commonly used for AC code implementation. AC implementation can be error prone. As the AC model is directly implemented by an algorithm, the errors are often caused by syntactic faults, such as mistakenly changing the + sign to – sign, or typing letter O instead of 0.

The correct implementation of the policy needs to be tested. To achieve that, a test oracle that contains cases of all possible outcome of the AC safety requirement is required, because implementation faults are unpredictable without a logical trace for detecting. Thus, all the combinations of the variables in the safety requirements need to be covered in the oracle. For example, a safety requirement: "*x read y grant*" where *x* has 3 different values and *y* has 5 different values will have 3*2*5*2 (assume that the AC actions has two values: *read* and *write*, and permission has only two values: *grant* and *deny*) test cases in the test oracle. The implemented AC system will then run these test cases to verify whether the actual test outputs are the same as the expected outputs.

It is not uncommon that a verification test includes hundreds of safety requirements; each contains tens of variables, in such case, the number of test cases in a test oracle for the implementation test is too great to be efficiently performed, therefor, additional techniques [8,9] for reducing the test case size without sacrificing the capability may be required for the test.

V. Implementation Considerations

General AC system testing framework shown in Fig. 8 contains four major functions using the methods as stated in Section IV. The AC Rule Real-time Error detector is used optionally for design the initial AC models [10]. The Black Box Tester checks if a model (original or mutant) holds for the specified Safety Requirements. The Black Box Tester (counterexample results) provides information for original model fix (a human action as dotted line in the Figure) and for mutation killing check for White Box Tester, it also takes output from the Test Generator and returns results for test case generation. The White Box Tester generates and kills mutant models based on the original model and safety requirements; its mutated models are sent to the Black Box Tester for

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

mutation killing check, or to the AC model author for original model or safety requirement fix (a human action as shown in dotted lines in the figure). The Test Oracle Generator generates test cases based on the Safety Requirement and the Black Box Tester's counterexample results. The process steps are listed below:

1. (optional) AC models is designed based on the AC policy by using AC Rule Real-time Error Checker (as described in Section 5.4).
2. Safety requirements are specified
3. Completed original AC model is checked against safety requirements by Black Box Tester, if an error is found, the original model needs to be fixed (thus repeat steps 1 to 3), otherwise proceed to next step 4.
4. Fixed original model send to White Box Tester for coverage and confinement check. The White Box Tester uses Black Box Tester to decide if generated mutant models were killed. If not, original model or safety requirements need to be fixed (thus repeat step 1 to 4), otherwise, proceeds to next step 5.
5. Test Oracle Generator generates test cases based on Safety Requirements, which are sent to Black Box Tester for generating permission results used for test oracle.

Note that the components in Fig. 8 and steps are not necessarily all required for an AC model verification; the selections of components and steps might depend on the complexity of the model and the cost for implementing the test framework. Thus, an AC model test framework can contain optional components/functions in Fig. 8 except that the Black Box tested is essential.

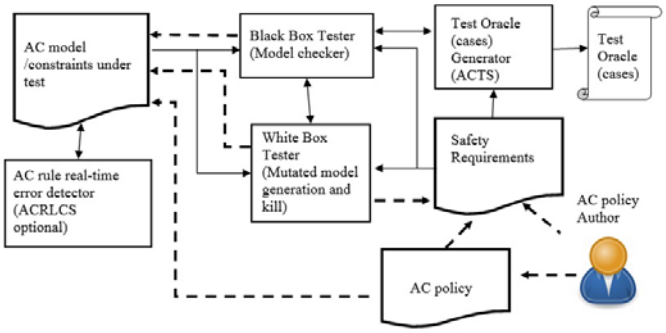


Fig. 8 AC model verification framework

VI. Related Work

In addition to the FSM based method, other techniques [11] are available for AC model verification such as theorem Proof (including first and higher logic proof) and MTBDD [12] methods.

Multi-Terminal Binary Decision Diagrams (MTBDD)

Developed in Racket (formal PLT) Scheme, Margrave [13] is a software tool suite for verifying safety requirements against AC policies written in XACML. Margrave represents XACML policies as MTBDD models, it allows the user to specify various forms of safety requirements in the Scheme programming language. Margrave uses one variable for each attribute-value pair in the XACML policy. Margrave creates MTBDD models for the individual policy rules, then combines these with MTBDD-combining algorithms that implement the XACML rule- and policy-combining algorithms.

Margrave views the policy constants permit and deny as rules; an operation called augment-rule takes a Boolean condition on the variables and a rule and constrains the rule to also require the given condition. It supports query-based verification and provides query-based views by computing exhaustive sets of scenarios that yield different results including change-impact analysis for comparing a pair of policies. Margrave provides the benefits of static verification without requiring authors to write formal properties; its power comes from choosing an appropriate policy model in first-order logic, and embracing both scenario-finding and multi-level policy-reasoning. In general, Margrave identifies formulas corresponding to many common firewall-analysis problems automatically, thus providing exhaustive analysis for richer policies and queries.

ACPT

NIST's Access Control Policy Tool (ACPT) [14] provides (1) GUI templates for composing AC models, (2) safety requirements verification for AC models through an SMV (Symbolic Model Verification) model checker NuSMV, (3) complete test cases generated by NIST's combinatorial testing tool ACTS, and (4) XACML policy generation as output of verified model. Through the four major functions, ACPT performs all the syntactic and semantic verifications as well as the interface for composing and combining AC models for AC policies; ACPT is capable of verifying combined policies based on the permission priorities and/or algorithms specified by the user.

ACPT allows users to specify AC models or their combinations, as well as safety requirements through GUI that contains model templates for three major AC policies: static Attribute-Based AC, Multi-Leveled Security, and stated Work-Flow. ACPT then performs black box model check to verify if the specified safety requirements conform to the specified models. If not, non-conformance messages are returned to the user, otherwise, ACPT proceeds to generate test cases through ACTS, which are ready for testing the AC application implemented according to the models.

Formal Methods

Formal methods for the validation of access control policies involving mathematical tools and proofs have also been advocated. Rémi Delmas and Thomas Polacsek [15] have proposed a logical modelling framework to find the inconsistencies and incompleteness in access control policies. Providing a mechanism for the detection of these two properties, they have introduced two new properties, applicability and minimality and their proposed technique is capable verifying these two properties [16]. By using the concepts of signatures, formula and predicates, they have defined some rules for the logical framework, which works for limited or finite data so their rules are also applicable to the finite data. They also mentioned that the MSFOL (many-sorted

first order logic) [17] formula should be converted to a pseudo-Boolean logic formula to analyze it. The proposed tool is a three-steps procedure where grounding operation gives the grounded formula in the first step which is converted to a bit-vector expression using the bit-vector encoding in the second step of this process. In the last step of this procedure, the bit-vector expressions are converted into clauses which are in pseudo-Boolean form and give us the pseudo-Boolean formula.

Z [18] is based on axiomatic set theory and first order predicate logic, which can be used for describing and modeling AC policies [19]. Z notation apply set theory forms an adequate basis for building the AC model, which allow syntax and type checking, schema expansion, precondition calculation, domain checking, and general theorem proving for model verification by domain checking. Many of the proof obligations are easily proven. In more difficult cases, generating the proof obligation is often a substantial aid in determining whether a specification in the AC model is meaningful to the AC policy.

VII. Conclusions

This paper describes a notion of safety for access control, and analyzes verification approaches for static, dynamic, and historical AC models. Static models are those in which no state is retained, while dynamic models may retain state during a session. Historical models include long-term user and object history in access decisions. An AC system is safe if no privilege can be escalated to unauthorized principals, but the correct privileges are always accessible to authorized principals. We also describe a rough taxonomy of faults that may be present access control models.

To verify safety requirements for AC models, we provided a general approach that expresses AC models and AC safety requirements in the formal specification of a black box model or first order logic checkers for verification. Then the black box verifier verifies the specified models against the specified safety requirements. Most of the verification system supports static, dynamic, and historical AC models. In addition to black box checking, white box checking methods make sure that the semantic coverage of the safety requirements also conforms to the intentions of the AC policy authors. Finally, the generation of test cases to check the conformance of the models and their implementations is necessary.

Access Control model and safety requirements conformance verification of generic AC policies bring benefits to society in two aspects. First, it should lead to improved verification practices for testing and verifying AC models in improving AC system quality and security in general. Second, innovations in new testing and verification algorithms and tools tend to propagate quickly across application or task domains where AC policies are used.

References

[1] V. C. Hu et al, NIST Special Publication 800-162, “Attribute Based Access Control Definition and Consideration.”

[2] V. C. Hu, D. F. Ferraiolo, D. R. Kuhn, NIST Interagency Report 7316, “Assessment of Access Control Systems”.

[3] V. C. Hu, and K. Scarfone, NIST Interagency Report 800-7874 “Guidelines for Access Control System Evaluation Metrics”.

[4] A. Gouglidis, I. Mavridis, V. C. Hu, “Security policy verification for multi-domains in Cloud systems” International Journal of Information Security (IJIS13), article No. s10207-013-0205-x, 13(2), 97-111 in Springer, July, 2014.

[5] Harriosn M. A., Ruzzo W. L., and Ullman J. D., “Protection in Operating Systems”, Communications of the ACM, Volume 19, 1976.

[6] V. Hu, R. Kuhn, T. Xie, and J. Hwang, “Model Checking for Verification of Mandatory Access Control Models and Properties,” International Journal of Software Engineering and Knowledge Engineering (IJSEKE) regular issue IJSEKE Vol. 21, No. 1., 2011.

[7] XACML, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[8] http://csrc.nist.gov/groups/SNS/acts/index.html

[9] V. C. Hu, R. D. Kuhn, T. Xie, “Property Verification for Generic Access Control Models”, in Proceeding of The 2008 IEEE/IFIP International Symposium on Trust, Security and Privacy for Pervasive Application (TSP2008), Shanghai, China, December 17-20 2008.

[10] V. Hu, K. Scarfone, “Real-Time Access Control Rule Fault Detection Using a Simulated Logic Circuit”, Proceeding, 2013 ASE/IEEE International Conference on Privacy, Security, Risk and Trust, Washington D.C., USA September 8th-14th, 2013.

[11] A. Li, Q. Li, V. C. Hu, and J. Di “Evaluating the Capability and Performance of Access Control Policy Verification Tools”, Proceeding The Premier International Conference for Military Communications (MILCOM 2015), Tampa FL, August 17-24, 2015

[12] E. Clarke, M. Fujita, P. McGeer, J. Yang, and X. Zhao, “Multi-terminal binary decision diagrams: An efficient data structure for matrix representation’, International Workshop on Logic Synthesis, 1993.

[13] K. Fisler et al, “Verification and Change Impact Analysis of Access Control Policies”, Proceeding, 27th International Conference on Software Engineering (ICSE’05), Page 196-205, ACM, New York, NY, 2005.

[14] http://csrc.nist.gov/groups/SNS/acpt/index.html.

[15] R. Abassi, S. Fatmi, "An Automated Validation Method for Security Policies: the firewall case", The 4th Int. Conf. on Information Assurance and Security, 2008, pp. 291-294.

[16] M. Aqib, R. A. Shaikh. “Analysis and Comparison of Access Control Policies Validation Mechanisms”, I.J. Computer Network and Information Security, 2015, 1, 54-69.

[17] J. H. Gallier, “Logic for Computer Science: Foundations of Automatic Theorem Proving”, ch. 10, pp. 448‑476, Wiley, 1987.

[18] B. Potter, J. Sinclair, and D. Till, “An Introduction to Formal Specification and Z” Second Edition, by Prentice Hall International Series in Computer Science, 1996.

[19] V. C. Hu, “The Policy Machine For Universal Access Control”, Dissertation, Computer Science Department, University of Idaho, 2002.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Restricting Insider Access through Efficient Implementation of Multi-Policy Access Control Systems

Peter Mell James Shook Serban Gavrila

ABSTRACT

The American National Standards Organization has standardized an access control approach, Next Generation Access Control (NGAC), that enables simultaneous instantiation of multiple access control policies. For large complex enterprises this is critical to limiting the legally authorized access of insiders. However, the specifications describe the required access control capabilities but not the related algorithms. Existing reference implementations have inefficient algorithms and thus do not fully express the NGAC's ability to scale. For example, the primary NGAC reference implementation took several minutes to simply display the set of files accessible to a user on a moderately sized system. To solve this problem we provide a efficient algorithm, reducing the overall complexity from cubic to linear. Our other major contribution is to provide a novel mechanism for administrators and users to review allowed access rights. We provide an interface that appears to be a simple file directory hierarchy but in reality is an automatically generated structure abstracted from the underlying access control graph that works with any set of simultaneously instantiated access control policies. Our work thus provides the first efficient implementation of NGAC while enabling user privilege review through a novel visualization approach. It thereby enables the efficient simultaneous instantiation of multiple access control policies that is needed to limit insider access to information (and thereby limit information leakage).

1. INTRODUCTION

Most operating systems provide simple access control mechanisms that are focused on enabling users to specify which other users have access to their files (i.e., Discretionary Access Control (DAC) [13]). However, many other access control approaches exist that provide enhanced features, especially for enterprise environments. This includes capabilities relevant to particular paradigms (e.g., for scenarios with financial transactions, handling of classified data, and conflict of interest) as well as greater simplicity in administering access control at scale (e.g., Role Based Access Control (RBAC) [5]). However, methods to use multiple approaches within a single enterprise have been lacking, that can result in enterprises settling for using a single simple model (e.g., DAC).

This can result in restrictions on insider access being defined very loosely, increasing the risk of insiders having unnecessary access to sensitive information and sharing that information outside of the organization. To ensure that users don't inappropriately share data, enterprises may then resort to the costly and inefficient approach of separating different data types (e.g., military classification levels) into totally distinct and isolated networks. Alternately, they may accept the risk of data being leaked, which can have disastrous results (e.g., classified documents being made public).

The American National Standards Institute (ANSI) has addressed this problem by standardizing an access control approach, Next Generation Access Control (NGAC) [6]. The NGAC stems from and is in alignment with the Policy Machine (PM) [9], a research effort by the National Institute of Standards and Technology (NIST) to develop a general purpose Attribute Based Access Control (ABAC) framework [12]. The NGAC is designed to enable simultaneous instantiation of multiple access control policies. The specification describes what constitutes a valid implementation using set theoretic notation but does not provide implementation guidance. This approach then leaves room for multiple competing approaches and implementations. In this work we will explore how the inefficiencies of the existing approaches make it infeasible to tightly restrict user access to data in large enterprises and we will provide scalable replacement algorithms that solve this problem.

To do this, we improve upon the algorithms in the open source NGAC reference implementation provided by NIST [3] in order to test achievable efficiency and scalability. Among other increases in efficiency, ours is the first to provide quadratically bounded algorithms for the retrieval of the set of user accessible objects where the existing reference implementations take cubic time (we are linear in making an access control decision for a single object). Furthermore, our algorithms are bounded to operating only on the access control sub-graph pertinent to a particular user (not the entire enterprise access control graph). This work is thus the first to demonstrate that NGAC can be scalable through the use of efficient graph algorithms. This in turn makes it feasible to tightly restrict insider access to data through use of multiple types of access control policies (and thus limit the leakage of sensitive data).

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Gavrila, Serban; Mell, Peter; Shook, James.
"Restricting Insider Access through Efficient Implementation of Multi-Policy Access Control Systems."
Paper presented at 8th ACM Computer and Communications Security International Workshop on Managing Insider Security Threats, Vienna, Austria. October 24, 2016 - October 28, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

For our initial efforts, we took the initial NGAC proof-of-concept implementation created by NIST [3] and evaluated the runtime of the algorithms. These were created by a highly skilled programmer who implemented the access control operations from the specification as they were stated in set theory notation (using a direct translation approach). This resulted in a functional system but one that could only scale to a couple of hundred users. At this scale, it took several minutes for a user to visualize their set of available objects. A complexity analysis of the code revealed cubic algorithms, explaining the lack of scalability of the implementation. We also looked at the other publicly available implementation of the NGAC which is provided as open source by Medidata [2]. Here, we found a user privilege determination algorithm that spent unnecessary time processing parts of the graph not accessible to the relevant user as well as a slow cubic algorithm for retrieval of all objects available to the user. It appears that with both implementations, it was an attempt to directly implement the NGAC set theory definitions that yielded the inefficient algorithms.

To solve this problem, we took a graph theoretic view to design an efficient algorithm for access control determination. We started by transforming the NGAC set theory into a graph representation (this was straightforward as the specifications themselves often use graphs to illustrate examples). Unfortunately, the resultant graphs had unusual features and constraints (with five different types of nodes, each with its own semantics). Thus, the primary challenge was in how to apply standard graph algorithms to this representation. Our solutions in general was to use breadth first search (BFS) and a depth first search (DFS) variant that performs a type of topological sort as primitive operations to allow us to cascade information from one type of node to another and percolate that information through the graph until the final answers are determined. The resultant algorithm is linear. Furthermore, it is not linear in relation to the entire access control graph, but only to the portion of the graph relevant to a particular user. This can offer even greater speedups, avoiding the need to even traverse the entire graph.

Besides not providing algorithms for calculating access control decisions, the NGAC standard does not provide any guidance on visualizing access control results to allow review of user privileges (see [11] and [15] for evidence of how this 'before the fact audit' capability is critically important). We presume the reason they do not provide this capability is because each access control policy may have its own preferred method for administrative review and user interaction. However, such a policy oriented approach isn't ideal in a system that simultaneously implements multiple policies (which is the whole point of NGAC). For example, the existing NIST PM approach requires users to choose a particular access control policy first and then navigate just within that policy structure to review user access (requiring the administrators and users to be knowledgeable about each access control policy and which files are covered by which policies). Because of these problems, there exists a need for a generic approach for user rights visualization that will work for any set of policies that can be instantiated within NGAC (without the staff having to understand said policies). Furthermore, this default visualization would ideally be automatically generated from the existing access control graph to avoid additional and excessive administrative burden.

To meet this need we provide the user (or the person reviewing the user's privileges) the visual experience of traversing a typical file directory hierarchy, as used by most major operating systems. However, under the hood the user is actually traversing the NGAC access control graph. We leverage one of the graph node types (object attributes) to act as file 'directories' enabling users to access their files. The user visually sees a tree but is actually traversing a graph with an exponential number of possible paths (where we generate local views on demand to avoid exponential calculations). Since the directory tree is automatically generated from the underlying graph, it can thus provide default user access to files simultaneously protected by multiple access control policies. An interesting side effect of our approach is that there can be multiple ways for a user to access the same file, without the need to explicitly create symbolic links. Thus, a document can be both stored under a person's personal directory and under a project directory with no duplication, system inconsistency, or need to explicitly create virtual links.

NGAC is not the only multi-policy access control system available. The current market leader appears to be the XACML standard [14] from OASIS [4]. Others include $ABAC_\alpha$ [12], HGABAC [16], and ABAC for Web Services [18]. The market leader XACML has been shown empirically to lack scalability in [17] where 3 different XACML implementations all experienced performance problems as the number of policies was increased past 100 (each policy in XACML contains the access rules for a set of target objects). This is not surprising as all of these logic-based formula approaches have been shown to be NP-complete with respect to enabling an administrator to review who can perform what actions. This is because policy review for such systems maps to the satisfiability problem [7]. This makes them undesirable for large enterprise systems with respect to ensuring the restrictions on insider access to sensitive data to avoid information leakage by insiders.

In summary, the contributions of this paper include:

1. the first ever study demonstrating the scalability of the NGAC multi-policy access control system,
2. a linear time algorithm for determining the objects and associated operations available to a user,
3. a novel visualization approach to enable review of user object access on NGAC systems,
4. and the ability for enterprises to efficiently implement multiple access control policies (which can limit insider access to information and thereby limit information leakage).

The remainder of this paper is structured as follows. Section 2 provides an overview of access control graphs within NGAC and provides a definition of when a user is allowed to access an object. Section 3 presents our access control algorithms and section 4 presents our visualization approach. Section 5 discusses related work and section 6 concludes.

2. ACCESS CONTROL GRAPH OVERVIEW

Using the NGAC specification [6] set theoretic definitions, we can form access control graphs as follows. There are 5 types of nodes to be created: user (u), object (o), user

Gavrila, Serban; Mell, Peter; Shook, James.
"Restricting Insider Access through Efficient Implementation of Multi-Policy Access Control Systems."
Paper presented at 8th ACM Computer and Communications Security International Workshop on Managing Insider Security Threats,
Vienna, Austria. October 24, 2016 - October 28, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

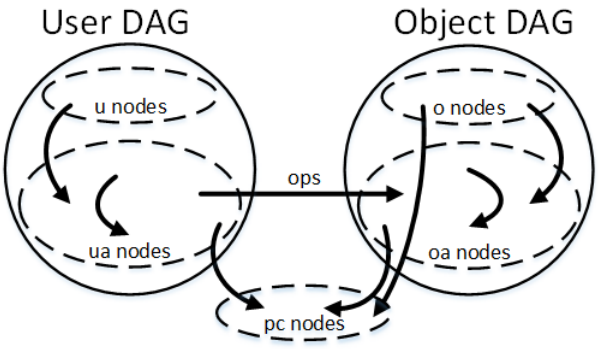


Figure 1: Diagram showing allowed edge relationships between the five different sets of NGAC node types

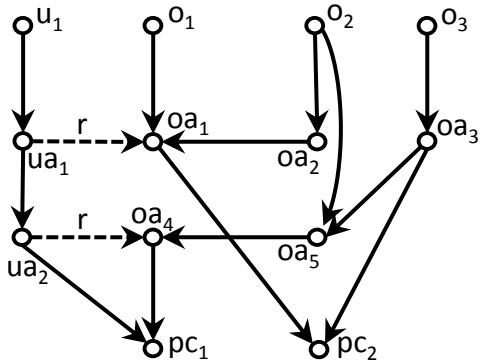


Figure 2: Example NGAC access control graph.

attribute (ua), object attribute (oa), and policy class (pc). All edges are directed. u nodes are sources with edges to ua nodes. ua nodes may have edges to ua, oa, and pc nodes[1]. oa nodes may have edges to oa and pc nodes. o nodes may have edges to oa and pc nodes. pc nodes are sinks. Cycles and self-loops are prohibited. ua to oa edges are labeled with a set of one or more allowed operations (ops) (e.g., read or write). All other edges are unlabeled. All nodes must have a path to at least one pc node (without using any ua→oa edges). For complexity evaluation purposes, the number of u, o, ua, and oa nodes are unbounded. However, the number of pc nodes and the number of distinct ops are assumed to be small constants.

These connectivity restrictions result in several features that we can leverage. The overall graph is a directed acyclic graph (DAG) that can be divided into two DAGs: a user DAG (with u and ua nodes) and an object DAG (with o and oa nodes). The set of u nodes act as sources for the user DAG and the set of o nodes act as sources for the object DAG. The set of ua to oa edges bridge the two DAGs and this bridge is the only place where edges are labeled, with operations (ops). We refer to the set of nodes on either side of these bridging edges as border nodes. The set of pc nodes act as sinks for both DAGs. The resulting overall graph is weakly connected.

An arbitrary access control graph can now be represented as shown in figure 1. Arrows within a set represent that nodes of that type can have edges to other nodes of that type, with no cycles allowed. This means that there are no edges between nodes within the set of pc nodes (the same is true for the set of u nodes and the set of o nodes). The arrows from the set of ua nodes to the oa nodes nodes represent the bridge edges (they contain the ops labels and connect the user and object DAGs). The bridge edges are the focal point in determining user privileges (see definition 1 below).

We now discuss how to determine user privileges. The ANSI NGAC standard provides set theoretic notation to enable computation of privileges abstracted away from any particular implementation. In this work, we describe the methodology using a graph oriented approach. Our graph theoretic derivation of the ANSI NGAC set theoretic definition of how to calculate access control is as follows[2]:

DEFINITION 1. *For a user, u_1, to perform an operation, op_1, on some object, o_1, there must exist a set of ua to oa edges with label op_1 such that the tail of each edge is reachable from u_1 and the head of each edge is reachable from o_1 and where the set of pc nodes reachable from the set of head nodes is a superset of the set of pc nodes reachable from o_1.*

Figure 2 shows an example NGAC access control graph which we will evaluate using definition 1[3]. Note that the dashed edges represent the bridge edges that connect the user DAG to the object DAG. The edge label 'r' represents read access. In this figure, user u_1 can read o_1 and o_2 but not o_3:

- o_1 requires pc_2 because there is a path connecting the two. This requirement is fulfilled by the edge $ua_1 \rightarrow oa_1$ providing 'read' access(because ua_1 is reachable from u_1, oa_1 is reachable from o_1, and pc_2 is reachable from oa_1). Thus by definition 1, u_1 can read o_1.

- o_2 requires pc_1 and pc_2 because there is a path connecting o_2 with both pc nodes. This requirement is fulfilled by a combination of the edges $ua_2 \rightarrow oa_4$ (which covers the pc_1 requirement) and $ua_1 \rightarrow oa_1$ (which covers the pc_2 requirement). Note that these two bridge edges would not have fulfilled the requirements had they different labels. Thus by definition 1, u_1 can read o_2.

- o_3 requires pc_1 and pc_2 because there is a path connecting o_3 with both pc nodes. Edge $ua_2 \rightarrow oa_4$ covers the pc_1 requirement for o_3. However, there does not exist a ua to oa edge that will satisfy o_3's requirement to cover pc_2. Edge $ua_1 \rightarrow oa_1$ does not work because oa_1 is not reachable from o_3 (which is required in definition 1). Thus by definition 1, u_1 cannot read o_3.

[1]In the NGAC specification, ua→pc edges are allowed but are not used for access control decisions.

[2]We don't include the NGAC definitions here because they use completely different set theoretic notation that would require extensive explanation and that is available in the NGAC standard).

[3]The edge $ua_2 \rightarrow pc_1$ fulfills the requirement in the specification that all u and ua nodes have a path to a pc node (without using bridge edges). However, the edge is not used for determining user privileges and will not be discussed further.

Gavrila, Serban; Mell, Peter; Shook, James.
"Restricting Insider Access through Efficient Implementation of Multi-Policy Access Control Systems."
Paper presented at 8th ACM Computer and Communications Security International Workshop on Managing Insider Security Threats, Vienna, Austria. October 24, 2016 - October 28, 2016.

u_1 = 'Bob'
ua_1= 'Bob Privileges'
ua_2= 'Death Star Personnel'
o_1 = 'Tatooine Vacation'
o_2 = 'Defense Systems Finances'
o_3 = 'Energy Shield'
oa_1 = 'Bob Personal'
oa_2 = 'Bob Deathstar Files'
oa_3 = 'Technical Designs'
oa_4 = 'Deathstar Project'
oa_5 = 'Defense Systems'
pc_1 = 'Access Control System 1'
pc_2 = 'Access Control System 2'

Table 1: Node Labels for Access Control Graph in Figure 2

Throughout this paper, we will use Figure 2 as an example where an accountant, Bob, is working on the defense systems finances for a deathstar. In this context, we can interpret the nodes as shown in Table 1. Note how the user attribute nodes represent 'teams' to which Bob belongs (his own team[4] and the death star personnel team) and the object attribute nodes provide a kind of hierarchy for different projects. In this example Bob has access to his own data (oa_1, oa_2, and o_1) as well as the overall project and financial material (oa_4, oa_5, and o_2). However, Bob does not have access to any of the technical designs (oa_3 and o_3).

3. ACCESS CONTROL ALGORITHMS

We now provide a linear time complexity graph algorithm to answer two of the most common types of access control requests: 1) is user, u_1, allowed to perform operation, op_1, on object, o_1 and 2) what is the set of accessible objects for a user, u_1, and what operations can u_1 perform on each object. Both of these determination can be made through a slight variation on the same algorithm, which we refer to as 'Fastg'.

3.1 The Fastg Algorithm

The Fastg algorithm first isolates the problem to just the object DAG through labeling each border oa node reachable from u_1 with a set of operations (from the ops labels on the bridge edges). Then, the set of objects of 'interest' are found by performing a reverse BFS from the set of reachable border oa nodes (without traversing any bridge edges). If we are simply trying to determine if u_1 can access a particular object, we intersect this object with the set of objects of interest (forming a new set of objects of interest). Finally, we perform a DFS from each object of interest and percolate up through the graph the set of reachable pc nodes and the set of reachable operations. Finally, for each reachable object, we compare the set of operations associated with each reachable pc node to determine if any operations are valid. For an operation to be valid it must be associated with all the pc nodes reachable from the o node.

In more detail, the algorithm is as follows:

1. BFS from u_1 to identify the set reachable ua border nodes (do not traverse oa nodes). For this set of ua border nodes, let the set of 'active' edges be the ua→oa outedges.
2. For each 'active' edge, label the oa head node with the ops edge label (eliminating duplicates). At this point, each reachable border oa node is labeled with a set of access rights.
3. Create a temporary node that is a successor of each reachable border oa node
4. Perform a backwards BFS from the temporary node (traversing edges in reverse) to find the set of objects of 'interest'. Do not traverse any bridge edges. Once done, delete the temporary node.
5. If the goal is to determine if u_1 can access a specific object, then intersect this object with the set of objects of interest to form a new set of objects of interest (this set will contain either a single node or be the empty set).
6. For each object of interest, perform a DFS to find the reachable pc nodes. However, when performing a DFS, label all nodes with the information found such that subsequent DFSs can take advantage of the previously computed information. Each object of interest then is labeled with its set of reachable pc nodes. These represent 'required' pc nodes for each object. Note that to record on each node which pc nodes are reachable, we need to modify the traditional DFS such that we only process a node (record the set of reachable pc nodes) if all of its successors have been processed. If a node with unprocessed children is pulled off the stack, we must put it back on the stack. The second time it is pulled off, it will be guaranteed that its children will all be processed.
7. While performing the modified DFSs from the previous step, perform an additional data propagation. When a reachable border oa node is labeled with its reachable pc nodes, associate those pc nodes with the operation labels from step 2. Then use the normal operations of the DFS to propagate these pc/operation pairings up to the root of the tree (one of the objects of interest). We use the same 'trick' from the previous step to reuse information between DFSs.
8. For each object of interest, compare the set of required pc nodes against the pc/operation pairings. If for some object, o_1, an operation, ops_1, exists that is associated with each required pc node, then u_1 is allowed to perform ops_1 on o_1 per definition 1.

The algorithm is apparently quadratic because we may perform a DFS from each object node. However, in steps 6 and 7 we store DFS results at each processed node such that the information can be reused by other DFSs. As a result, the set of executed DFSs is guaranteed to traverse each edge in the object DAG at most twice. The BFS from step 1 traverses each edge in the user DAG once and each bridge edge once. Step 2 traverses each bridge edge once. And step 4 traverses each object DAG edge at most once. In summation, each edge in the graph is then guaranteed to

[4]We had to create ua_1 to represent Bob's access rights because the NGAC specification does not allow creation of u to oa edges.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Number of user nodes = $.1 * n$
Number of user attribute nodes = $.1 * n$
Number of object nodes = $.5 * n$
Number of object attribute nodes = $.3 * n$
Number of pc nodes = 3

Table 2: Proportion of Nodes of Each Type

be traversed at most 3 times (most much less and some not at all). This makes the algorithm linear with respect to the number of edges, O(m).

3.2 Empirical Algorithm Results

In this section, we evaluate the scalability of our Fastg algorithm versus the two available reference implementations (NIST PM and Medidata). For our experimental platform we used an Ubuntu virtual machine with two cores and 10 Gb of memory running on a commodity laptop. For software to encode the algorithms, we used Python 2.7 and NetworkX (a graph algorithms library). Faster execution times can be achieved through use of more efficient programming languages (e.g., C) but our goal is to evaluate relative performance of the algorithms. These results then are an upper bound on what can be achieved relative to execution time. With respect to memory, none of the algorithms used even a majority of the available memory and thus we do not report memory usage statistics.

For our empirical scalability study, we used the Fastg variant that computes the set of accessible objects for a particular user and the set of operations available for each object. For comparitive purposes, we coded up the analogous algorithms from both NGAC reference implementations (the NIST PM [3] and Medidata [2]) using the same language and libraries. These implementations are discussed in 1 and then further in 5. The Medidata algorithm was perfectly analogous (identical inputs and outputs), however the NIST PM algorithm performed additional work not required to obtain our desired output. For example, the NIST PM algorithm outputs the oa nodes accessible to a user, not just the o nodes. To avoid unfairly penalizing the NIST PM algorithm, we included in our implementation only those parts relevant to obtaining the desired output.

To test the scalability of the algorithms, we generated access control graphs that varied in size from 1000 to 700,000 nodes. We used the number of nodes,n, as the independent variable and then scaled all other graph features relative to n. The proportion of nodes of each type (u, ua o, ou, and pc) are shown in table 2. For edges, we calculated an Erdos-Renyi edge probability, p, used to create random graphs [8] such that the mean number of edges per node would be no more than 5. Then, for each candidate edge allowed by the NGAC specification, we used p to determine whether or not to place the candidate edge in the graph. The only exception is that we limited the length of the u to pc paths in the user DAG and o to pc paths in the object DAG to be at most 5. We did this for the user DAG by dividing the ua nodes into 4 groups labeled with consecutive integers. Edges leaving a node were only allowed to go to nodes in groups with higher labels (edges within a group were not allowed). A similar operation was performed for the object DAG.

There do not exist any references that one can leverage for creating random NGAC graphs. Thus, we assigned the above parameters according to qualitative expert domain knowledge to create as realistic NGAC graphs as possible. To make sure that any particular parameter choice did not unfairly hamper one of the algorithms, we ran numerous experiments (not shown) where for a graph size of 2000 nodes, we varied the following parameters: proportion of nodes of a particular type (u, ua, o, oa, and pc), number of layers for the user and object DAG (to include turning off this feature), and the mean number of edges per node. We chose graphs of 2000 nodes for this experiment because that was the maximum size at which all three algorithms had a less than 20 second execution time. Some of these parameter changes produced no significant effect on execution time (e.g., number of pc nodes) while others produced significant changes (e.g., those related to the number of edges in the graph). The number of edges in the graph was affected by two factors: the number of candidate edges and the p variable. The number of candidate edges was changed by varying the proportion of ua and oa nodes and the number of layers. The p variable used to calculate whether or not to instantiate a candidate edge was changed by varying the parameter for the mean number of edges per node. While we were able to change the execution times through parameter manipulation, the relative execution times between the three algorithms remained the same.

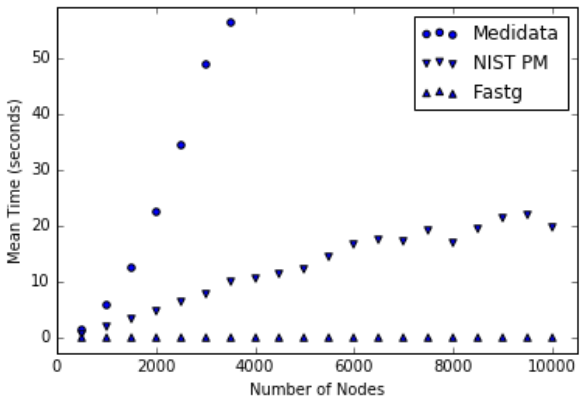


Figure 3: Execution time on graphs up to 10,000 nodes

In the figures, we refer to our algorithm as 'Fastg' and the other two as 'Medidata' and 'NIST PM'. For each data point, we took the mean of 300 trials. We limited each algorithm to taking no more than 60 seconds, at which point we terminated further use of that algorithm. In an actual NGAC deployment, the required response time to show a user their accessible objects is more likely to be less than 2 seconds.

Figure 3 shows the timing for all three algorithms for graphs up to 10,000 nodes. At 10,000, the Fastg algorithm took a mean of .077 seconds to retrieve the set of objects available to a particular user. The NIST PM algorithm was 285 times slower, taking 22 seconds. The Medidata algorithm exceeded the 60 second limit at just 4000 nodes. Given that the required response time in an actual deployment is likely just a couple of seconds, the Medidata and NIST PM algorithms are limited to being used on graphs

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Gavrila, Serban; Mell, Peter; Shook, James.
"Restricting Insider Access through Efficient Implementation of Multi-Policy Access Control Systems."
Paper presented at 8th ACM Computer and Communications Security International Workshop on Managing Insider Security Threats,
Vienna, Austria. October 24, 2016 - October 28, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

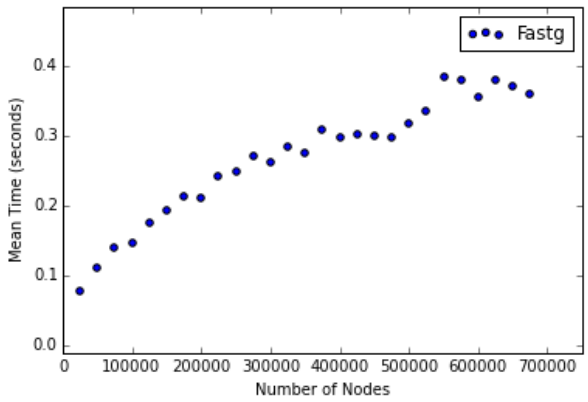


Figure 4: Execution time on graphs up to 700,000 nodes

with less than a couple of thousand nodes (that conform to our parameters).

Figure 4 shows the performance of the Fastg algorithm on graphs up to 700,000 nodes. The Fastg algorithm takes less than .4 seconds at 700,000 nodes. Given our assumed operation requirements of less than 2 seconds, this makes Fastg scalable up to the largest graphs that we produced (that conform to our parameters). We did not generate larger graphs due to execution time and memory limitations on our code used to produce the NGAC graphs.

Note, great care must be taken in interpreting these results. Our intention was to create as realistic graphs as possible and then show that the relative performance of the Fastg greatly outperformed that of the Medidata and NIST PM solutions. In this work, we have done that both theoretically and, in this section, empirically. However, NGAC graphs from operational deployments may have different parameter values or properties not modeled by our graph simulator. Such differences can greatly effect the absolute timing values (as we saw in our experiments on graph of 2000 nodes in changing the parameter values). Thus, we caution the reader to avoid using this work to calculate a precise upper bound on the size of graph that can be processed by any of the three algorithms. That said, the linear nature of Fastg should make it suitable for use on most any realistic NGAC graph.

4. ACCESS CONTROL VISUALIZATION

These graph algorithms enable access control decisions to be made while simultaneously instantiating multiple access control policies. However, a major question remaining is how to effectively communicate this set of privileges to the users. To this end we have designed an access control visualization approach that meets the following goals:

1. Leverage the access control graph to create a default visualization method for review of user file access

2. Abstract away the access control policy details such that the users (or administrators) do not need to understand the policies nor need to know which of the files are covered by which policies

Meeting these goals will enable efficient review of user privileges to best limit insider access to information (and thereby limit information leakage).

The NIST PM implementation, version 1.5, meets the first goal by leveraging the access control graph. This approach uses the PM itself as a root node in a file hierarchy and then the instantiated access control policies as the second level folders. Clicking on the access control policies enables the user to traverse the object DAG backwards (displaying only oa nodes pertaining to the chosen policy) until reaching the desired files. Unfortunately, this approach does not meet our second goal because their system requires users to navigate to their files by knowing which files are covered by which policies.

Our solution is to use the user's name as the root in a hierarchical file structure. The second level 'folders' are the labels for the border oa nodes reachable from the user's u node. Given the importance of the border edges in the NGAC access control definition 1, it is natural to use the border oa nodes as the first layer of file organization for the user. When a user clicks on an oa node name, the next level folders that appear are the oa node predecessors in the object DAG for which the user has some privilege. This graph traversal stops whenever the user reaches the object leaf nodes.

In our approach, we abstract away the complexity of the access control graph to make it appear to the user as if they are traversing the usual hierarchical directory structure used by default in all major operating system. In reality, the user is traversing possibly overlapping paths of the graph. The number of such paths is exponential and so we perform calculations only on the path actually being traversed by the user. Furthermore, there may be multiple ways for a user to access a particular file. This enables built in flexibility that previously had to be provided explicitly with artifacts such as symbolic links.

Figure 5 provides an example view of a user's accessible objects taken from one of our testing datasets covering a medical scenario. While it appears to be a typical file hierarchy, note how there are multiple paths by which to traverse to particular files (demonstrating that we are actually traversing a graph). For example, file 'mrec1' is available via three different paths in the graph: $root \rightarrow TS$, $root \rightarrow MedRecords$, $root \rightarrow alicehome \rightarrow AliceMedRecords$. In fact, all files shown in this visualization depict this multi-path behavior except for the files 'DAC uattrs rep' and 'alice home rep'.

4.1 Predecessor Node Visualization Algorithm

We now provide an efficient algorithm to determine what files and folders to show when a user clicks on some 'folder'. Initially, this will be one of the labels for the border oa nodes reachable from the user node, u_1. The algorithm is as follows:

1. Let the 'folder' on which the user clicks correspond to an oa node, x (note that this algorithm assumes that x is a folder that u_1 has the ability to view). Find the 'covered' pc nodes for x by performing a BFS and including all reachable pc nodes.

2. Find the set of predecessors of x and for each predecessor node, y, find the required pc nodes by performing a BFS and including all reachable pc nodes.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

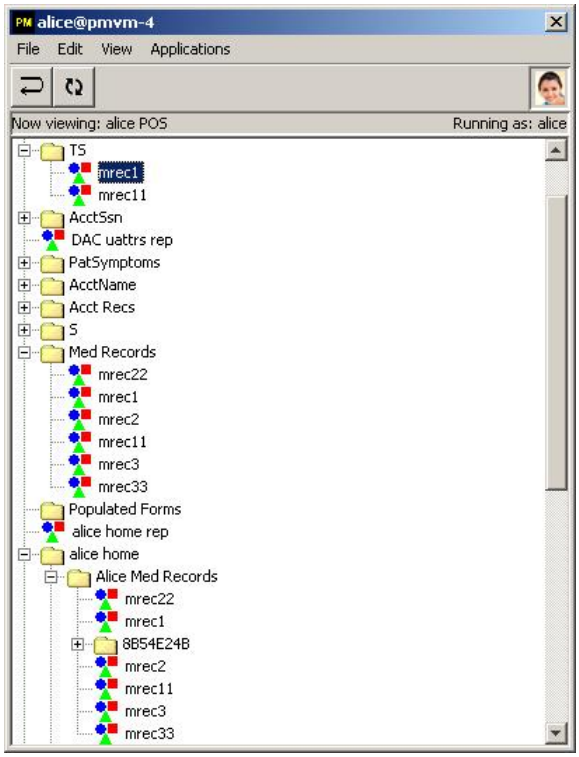


Figure 5: Example hierarchical visualization of a user access rights directed acyclic graph

3. For each predecessor node, y, if the set of required pc nodes is equal to the covered pc nodes for x then add it to a list of nodes available for display. If a node doesn't make it on this list in this step it doesn't meant that u_1 can't access it (simply we currently don't have enough evidence).

4. If there are predecessor nodes not on the list of nodes available for display, execute the 'Border oa Labeling' algorithm described below for u_1.

5. For each predecessor node, y, not on the available node list, perform a BFS from y to find all labeled border oa nodes (from the previous step). Let the 'available rights' for y be the union of the access right/pc node pairings from these reachable labeled border oa nodes. From this set of pairings, create a hash table where the keys are the access rights and the values the set of pc nodes. If there exists any key for which the values are a superset of the required pc nodes for y, then add y to the list of nodes available for display.

The 'Border oa Labeling' algorithm used above in step 4 is as follows:

1. BFS from u_1 to identify the set reachable ua border nodes. For this set of ua border nodes, let the set of 'active' edges be the ua→oa outedges.

2. For each 'active' edge, label the oa head node with the ops edge label (eliminating duplicates). At this point, each reachable border oa node is labeled with a set of access rights.

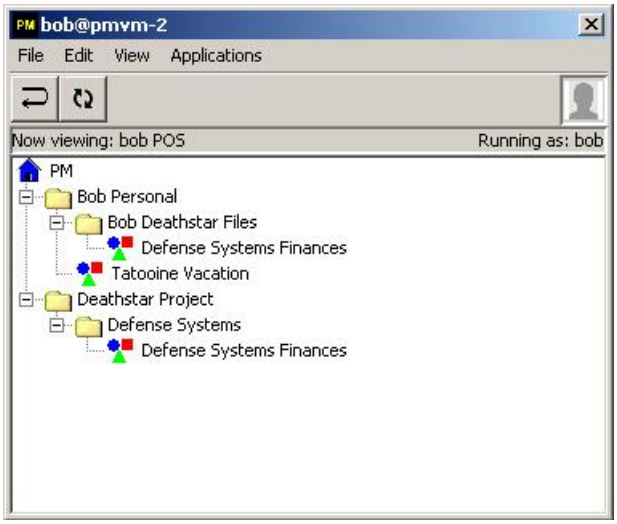


Figure 6: Example File Hierarchy for Access Control Graph in Figure 2

3. From each pc node, perform a backwards BFS (traversing edges backwards) to find labeled oa border nodes. For each such node, label it with the set of reachable pc nodes.

4. Each processed oa border node is then labeled with the cross product of the union of the access right labels with the set of reachable pc nodes (forming the access right/pc node pairings).

The combination of these two algorithms is linear, $O(n+m)$ (assuming as usual that the number of distinct access right types and policy classes are a small constant).

4.2 Visualization Examples

We now return to our example of the accountant Bob who is working on the defense systems finances for a deathstar. We will use our visualization approach to show the files available to Bob in Figure 2 using the node labels from Table 1.

The fully available hierarchical tree for user Bob is shown in Figure 6. This assumes that Bob has clicked on the 'Bob Personal' folder followed by a click on the 'Bob Deathstar Files' subfolder. It also assumes that Bob has clicked on the 'Deathstar Project' folder followed by a click on the 'Defense Systems' folder. These four clicks expand out visually all of Bob's available folders and files as shown in Figure 6. Note that the 'Technical Designs' folder and the 'Energy Shield' file are not visible because they are not accessible to user Bob.

A feature of this approach is that user Bob has access to the 'Deathstar Finances' file through both his own documents folder as well as the 'Deathstar Project' folder (logically this is because Bob is the owner/maintainer of that file). This again demonstrates the power of the approach where the user visually sees a hierarchy but can access the same files through multiple paths (without the need to explicitly create such linkages).

Note that while Bob has access to the 'Deathstar Project' folder, he is unable to see anything regarding the 'Technical Designs' folder including the 'Energy Shield' file. For Bob to

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

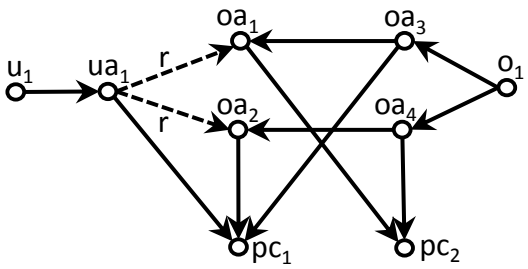


Figure 7: Minimal Access Control Graph Containing an Orphaned File

be able to access the 'Technical Designs' folder and 'Energy Shield' file, there would have to exist an edge $oa5 \rightarrow oa2$, $oa3 \rightarrow oa2$, $oa5 \rightarrow oa1$, or $oa3 \rightarrow oa1$ (see Figure 2). Alternately, the existence of an edge $o3 \rightarrow oa1$ or $o3 \rightarrow oa2$ would be sufficient to allow Bob access to the 'Energy Shield' file per Definition 1. However, for this our visualization approach would not allow Bob to use the 'Technical Designs' folder because it would still not be accessible to Bob. In this case, Bob could access the 'Energy Shield' file through the folder 'Bob Personal'. Thus, when a user can't get to one of their files through some particular oa node, there generally exists another oa node that will permit access through the visualization approach.

4.3 Orphan Files

However, there does exist the possibility that a user may not be able to traverse the visualization to reach a file that by Definition 1 is accessible. We call such files 'orphan' objects. None of the examples in the NGAC or the PM specification will generate orphans. Likewise, in our own test datasets we have never experienced an orphan file. Nevertheless, the possibility exists and so we discuss approaches to allowing for this eventuality.

For an orphan file to exist for a particular user, there must be an object node that is accessible but each path from the object to the set of reachable border oa nodes has a node that is not accessible. This happens when for each path, there exists an intermediate node that 'requires' a policy class not provided by the path's border oa node. An intermediate oa node requires a policy class when it has a path to that pc node (see Definition 1). Note that while the intermediate nodes on each path are not accessible, each path provides user privileges to the object such that the union of the received privileges enables the object to be accessible.

Figure 7 shows the simplest possible access control graph with an orphan file. o_1 is accessible because it receives pc_1 read privileges from oa_2 and pc_2 read privileges from oa_1. However, oa_3 is not accessible because it requires pc_1 privileges but only receives pc_2 privileges from oa_1. Likewise, oa_4 is not accessible because it requires pc_2 privileges but only receives pc_1 privileges from oa_2.

We have identified three different approaches to handling the possibility of orphan files such that the user can still find and access them (in order of increasing cost of computation time):

1. Enable the user to perform a search through all accessible files as a method to have access to any orphaned files. Our algorithm to find accessible objects (section 3.1) provides a list of all accessible files, both orphaned and available through the visualization approach. The user can simply perform a regular expression search on that list.

2. In the user's visualization of their file hierarchy, provide a folder at the second tier (alongside the reachable border oa node labels) that is labeled 'Orphan Files'. The orphan files can be detected when first launching the visualization and then listed in that directory. Our quadratic algorithm for finding orphan files is provided below.

3. Show the user orphaned files while they are traversing their hierarchical file structure. Whenever a non-accessible folder is encountered, perform a search for orphaned nodes only above the non-accessible folder. If orphans are encountered then show them in the current directory with a special designation to indicate that they are orphans. To the best of our knowledge, this approach requires executing our full quadratic 'find orphans' algorithm (below) followed by a reverse BFS to determine which orphans should map to the non-accessible folder.

4.4 Orphan Node Detection Algorithm

This algorithms enables one to detect orphan nodes for a user node u_1. The algorithm is as follows:

1. Execute the 'Border oa Labeling' algorithm to label the border oa nodes, reachable from u_1, with access right/pc node pairings (see section **??**).

2. Create a hash table where the keys will be node names and each value will be a set of access right/pc node pairings.

3. From each visible border oa node, x, BFS up (traversing the edges backwards) over the object DAG (i.e., don't traverse any ua→oa edges). For each visited node, y, add it to the hash table (if it isn't already there). Add x's access right/pc node pairings to the value set for y.

4. For each key in the hash table, x, perform a BFS down (traversing edges forwards) to find the required pc nodes. If the value set for x does not contain some privilege for which all required pc nodes are covered, then delete this key from the hash table. In more detail, the value set must have access right/pc node pairings with some privilege, p, where the associated pc nodes in the pairings with p must be a superset of the required pc nodes. The resulting hash table will contain only nodes that are accessible to u.

5. For each key, x, in the reduced hash table that references an o node (not an oa node), BFS down (traversing edges forwards) attempting to reach a visible border oa node (as identified previously). However, modify the BFS to only traverse nodes that are referenced as keys in the reduced hash table. Nodes not in the hash table are either not accessible to u or will not provide a path to one of the visible border oa nodes. If the BFS terminates without reaching any visible border oa node, add x to a list of orphaned objects.

Steps 3 and 5 cause the algorithm to be quadratic. Steps 4 is described as being quadratic (for clarity) but can be made linear if one initiates the required BFSs from the pc nodes. The overall algorithm is quadratic.

5. RELATED WORK

In this section, we discuss the NGAC reference implementations and algorithms that preceded this work. There are two public NGAC/PM reference implementations; both available on GitHub [10]. NIST provides a reference implementation in Java that was the primary reference used in the development of the NGAC [3]. The company Medidata provides an implementation in Ruby that they use for their software products in the medical field [2]. A third GitHub policy machine implementation is available from Colorado State, but we will not reference it further as it focuses on using the NIST PM implementation to manage application-level operating system resources in Linux environments [1].

For the NIST implementation, we evaluated version 1.5. Their code related to determining which resources are available to a particular user is cubic, which explains the slow execution time even on small test sets. We provided our algorithms to NIST PM development team and they plan to use a variant on our access control algorithm as well as our visualization approach in an upcoming software release.

For the Medidata code base, we evaluated their default implementation in the file '\lib\policy_machine.rm' of version 1.1.0. Note that they have alternate implementations that we did not analyze that use a graph database and relational database (they do not recommend use of the graph database as they claim the 'interface is slow' and we didn't have access to their relational database to test it). For their default 'in memory' implementation, they have an $O(nm^2)$ cubic execution time method 'accessible_objects' that determines which files a user can access. Their method 'is_privilege', to determine if a user has a specific privilege on a particular object, is also quadratic while ours is linear. We provided them our algorithms and they plan to use them to improve their default implementation.

It appears that both implementations are inefficient due to a direct translation of the set theoretic NGAC notation into computer code.

6. CONCLUSION

The lack of an efficient system to simultaneously instantiate security policies has resulted in the use of blunt mechanisms to restrict user access to data (e.g., reliance on just DAC and isolated networks for differing levels of data sensitivity). This has resulted in insiders having access to more data than is necessary to perform their job function, exacerbating the impact of insiders leaking sensitive information. The NGAC provides a solution to this important problem by enabling the instantiation of multiple security policies within a single access control system. It quite appropriately provides requirements without specifying implementation details, allowing for competing approaches. However, the existing reference implementations use cubic algorithms, which raised into question whether or not NGAC can be implemented efficiently. Furthermore, NGAC did not provide guidance on how to visualize the results of the systems, making it unclear how perform reviews and audits of user access.

This work addressed both of these issues. We provide the first implementation of NGAC using an efficient linear time algorithm (bounded to the parts of the graph relevant to the user). Furthermore, we provide a novel visualization approach that works by default with multiple access control policies and that enables efficient review of user access rights.

7. REFERENCES

[1] Colorado state 'tinypm' implementation on github.

[2] Medidata policy machine code on github, version 1.1.0.

[3] Nist policy machine code on github, version 1.5.

[4] Organization for the advancement of structured information standards (OASIS).

[5] ANSI. American national standard for information technology, role-based access control (RBAC). Technical Report ANSI INCITS 359-2004, American National Standards Institute, 2004.

[6] ANSI. Information technology - next generation access control - functional architecture (NGAC-FA). Technical Report ANSI-INCITS 499-2013, American National Standard Institute, 2013.

[7] P. Biswas, R. Sandhu, and R. Krishnan. Label-Based Access Control: An ABAC Model with Enumerated Authorization Policy. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, ABAC '16, pages 1–12, New York, NY, USA, 2016. ACM.

[8] B. Bollobas. *Random graphs*. Cambridge studies in advanced mathematics. Cambridge university press, Cambridge, New York (N. Y.), Melbourne, 2001.

[9] D. Ferraiolo, S. Gavrila, and W. Jansen. Policy machine: Features, architecture, and specification. Technical Report NISTIR 7987 Revision 1, National Institute of Standards and Technology, Oct. 2015.

[10] GitHub. Github code repository.

[11] V. Hu, D. Ferraiolo, and D. Kuhn. Assessment of Access Control Systems. Interagency report, National Institute of Standards and Technology (NIST), 2006.

[12] X. Jin, R. Krishnan, and R. Sandhu. *A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC*, pages 41–55. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[13] NCSC. *A Guide to Understanding Discretionary Access Control in Trusted Systems*. Number NCSC-TG-003. National Computer Security Center, Fort George G. Meade, Maryland, USA, 1 edition, Sept. 1987.

[14] OASIS. eXtensible access control markup language (XACML) Version 3.0., OASIS Standard, Jan. 2013.

[15] A. C. O'Connor and R. J. Loomis. 2010 Economic Analysis of Role-Based Access Control. Technical Report RTI Project Number 0211876, RTI International, 3040 Cornwallis Road Research Triangle Park, NC 27709, Dec. 2010.

[16] D. Servos and S. L. Osborn. *HGABAC: Towards a Formal Model of Hierarchical Attribute-Based Access Control*, pages 187–204. Springer International Publishing, Cham, 2015.

[17] F. Turkmen and B. Crispo. Performance evaluation of XACML PDP implementations. In *Proceedings of the 2008 ACM Workshop on Secure Web Services*, SWS '08, pages 37–44, New York, NY, USA, 2008. ACM.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

[18] E. Yuan and J. Tong. Attributed based access control (ABAC) for web services. In *IEEE International Conference on Web Services (ICWS'05)*, page 569, July 2005.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

DANE Trusted Email For Supply Chain Management

Joseph Gersch
Secure64 Software Corporation
joe.gersch@secure64.com

Dan Massey
Colorado State University
massey@cs.colostate.edu

Scott Rose
NIST
scott.rose@nist.gov

Abstract

Supply chain management is critically dependent on trusted email mechanisms that address forgery, confidentiality, and sender authenticity. The IETF protocol 'Domain Authentication of Named Entities' (DANE) described in this paper has been extended from its initial goal of providing TLS web site validation to also offer a foundation for globally scalable and interoperable email security. Widespread deployment of DANE will require more than raw technology standards, however. Workflow automation mechanisms will need to emerge in order to simplify the publishing and retrieval of cryptographic credentials that are applicable for general audiences. Security policy enforcement will also need to be addressed. This paper gives a descriptive tutorial of trusted email technologies, shows how DANE solves key distribution logistics, and then suggests desirable automation components that could accelerate deployment of DANE-based trusted email. Pilot deployments are briefly described.

1. Introduction

Email is one of the most critical communication tools used in supply chain management. It is relied upon for a wide range of messages: partner-to-partner, customer-to-vendor, order processing and billing, and everyday intra- and inter-company communications. The inconvenient truth, however, is that email as typically used today *cannot* be relied upon.

It is difficult to tell if an email is fraudulent. An original email message can be modified by a man-in-the-middle attack; for example, to alter a bank routing number used for electronic payments. Phishing and spear phishing attacks are common and have become extremely sophisticated. Attackers are able to manipulate organizations for financial gain, espionage, or to launch malware.

Email is the preferred channel for launching targeted cyber attacks. Email is the weak link in government and enterprise security; it is hard to protect because email is not secure and is subject to social engineering. There are numerous examples of the abuse of email. A sampling of reports sorted from 2011 to 2016 shows a growing trend to targeted spear phishing:

- The 2011 *OMB Report to Congress* cites US CERT (The United States Computer Emergency Readiness Team) reporting 51.2% of 107,655 incidents reported by public agencies were phishing [1, 2].
- The Cisco[1] 2011 Security White Paper *Email Attacks: This Time It's Personal* illustrates the economic gain for attackers in moving away from mass attack phishing to targeted spear phishing attacks. In just one year the cyber criminal monetary benefit rose from $50 million to $150 million [3].
- Trend Labs 2012 Research Paper *Spear-Phishing Email: Most Favored APT Attack Bait* indicates that 65% of incidents were targeted to Government [4].
- The 2016 Verizon *Data Breach Investigation Report* states that 30% of phishing messages were opened by targets and 12% went on to click malicious attachments. The majority of phishing cases are used as a means to install persistent malware. Cyber-Espionage was found in 68 examples of phishing/social engineering attacks [5].
- As a specific example, Arrow Electronics, a major distributor, revealed that they were the victims of a $13 million theft in early 2016 based on a combination of social engineering and spear

[1] Certain commercial equipment, instruments, or materials (or suppliers, or software,...) are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Gersch, Joseph; Massey, Daniel; Rose, Scott.
"DANE Trusted Email for Supply Chain Management."
Paper presented at Hawaii International Conference on System Sciences HICSS-50
Supply Chain Security and Mutual Trust Research Minitrack, Honolulu, HI. January 4, 2017 - January 7, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

phishing in which an executive was impersonated [6].

Various approaches have been used to mitigate these problems. Application firewalls, Bayesian spam filters, email gateways and portals are common examples. The core solution, however, is to employ the inherent trust mechanisms contained in the email protocol itself. Email should be automatically encrypted and digitally signed to ensure message integrity and sender authentication to eliminate spear phishing attacks.

Trust mechanisms for email have existed for decades, but unfortunately these remain mostly unused or misunderstood. Barriers to use include the lack of a globally scalable publishing & retrieval mechanism for end-user cryptographic certificates and the complexity of current email security solutions. Ease-of-use is a common objection from anyone who has set up or renewed personal email certificates in laptops and mobile devices. Automated policy enforcement is also lacking.

Extensions to the DNS-based DANE protocol have been published [7, 8, 9] to address improvements for email security. In a nutshell, these DANE extensions use the existing infrastructure of DNS and DNSSEC to create a secure global repository of end-user X.509 certificates and the cryptographic credentials that authenticate email servers.

By itself, however, it would be unlikely for DANE to be widely deployed for the same reasons that S/MIME is not widely used; the lack of simple-to-use end-user solutions. The current email security ecosystem has multiple interdependent components that involve PKI certificate authorities, DNS provisioning systems, email host servers, and email client programs that run in a variety of end-user devices such as laptops, tablets and smartphones. Consumers today are used to a world where entire solutions are available by simply “downloading an app”, not by having to integrate pieces from multiple sources using a complicated set of installation instructions from a variety of vendors.

The benefits of DANE will not be realized without catalyzing its technology within a broader approach for ease-of-use. Due to email’s history, evolution, and the wide variety of vendors, it is also unlikely that there will be a day-one event in which all components are simultaneously interrelated.

This paper suggests methods to overcome these usage barriers via an incremental approach towards ease of use. We advocate automation techniques to manage the provisioning, maintenance and policy directives for credentials. Each step is useful in its own right; combined together they bring us closer to a more complete and deployable solution for end consumers. We also describe current pilot implementations of DANE email extensions and proposed international government mandates.

The remainder of this paper is organized as follows. Section 2 gives background regarding basic email security mechanisms. Section 3 describes the IETF DANE email extensions. Section 4 suggests components for automation and ease-of-use that would enable wider deployment of DANE. Section 5 concludes the paper.

2. Background and Related Work

2.1. S/MIME and OpenPGP: Use and Limitations

The email protocol [10] is over 30 years old and was originally restricted to text-only messages. It was later enriched with Multipurpose Internet Mail Extensions (MIME) for attaching files, formatted text, HTML audio, video, applications and graphics [11, 12, 13]. This extended its usefulness beyond measure. Trust mechanisms for confidentiality, authentication and data integrity were addressed by extending email with Secure/MIME (S/MIME) [14, 15] and with an alternative method, OpenPGP for MIME [16]. Both S/MIME and OpenPGP use public key cryptography to digitally sign and encrypt email messages.

Public Key Cryptography is a method in which an email user generates a public/private key-pair that is either signed by a Certificate Authority (CA) or encoded into a self-signed certificate. The added value the CA brings is that it is a third party that is vouching for some portion of the identity metadata stored in the certificate along with the public key. The public key certificate is meant to be globally available to anyone so that they may use S/MIME to encrypt email. The private key, held only by the email recipient, is used to decrypt these messages. The private key is also used to generate digital signatures for email. Since only the sender has the private key, this mechanism ensures authenticity of the email sender and additionally ensures that no changes were made to the message (data integrity). Fraudulent email will not be able to be signed.

Unfortunately, use of S/MIME today is spotty at best. The trust mechanism is cryptographically sound, but operational issues have stalled its use. These include:

- Creation of user key-pairs and installation of private keys onto multiple devices.
- Global Distribution of public key certificates.
- Lack of a name-space to authenticate public key certificates

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

- Resisting spammer techniques such as ”cousin domains”
- Lack of enforced policy and feedback mechanisms

The manual steps involved in generating and installing personal cryptographic keys can be difficult and time-consuming, therefore most users simply don't do it. Furthermore, users have multiple devices and multiple email identities. Transporting private keys from a laptop to a smartphone or tablet is a possible but confusing process. The end result: no keys, no trusted email, increased risk.

Assuming a user has mastered the art of key installation and management, the next step is to distribute the public key certificates. Unfortunately, there is no global key repository in which one can publish and retrieve the public key of an individual. Instead, it is usually done by S/MIME users manually distributing keys to desired recipients by sending them a digitally signed email. OpenPGP distributes keys using a web of trust via ”key-exchange parties” and a limited set of well-known key exchange servers. Neither S/MIME nor OpenPGP scale well and this limits use. A vendor cannot send encrypted email to a customer for whom the key is unknown.

Another operational problem is the existence of fraudulent certificates. It is possible for rogue CAs to generate fake server or email certificates. Recipients don't normally examine email certificates to see if they are correct. They assume that if a certificate exists, it must be valid. To avoid using malicious credentials, it is desirable to link the authorized certificates into a global managed name space such as the Domain Name System (DNS). This is described further in the next section.

Related to fake certificates is the use of “cousin domains”, defined by Steve Crocker as “*a registered domain name that is deceptively similar to a target domain name. The target domain is familiar to many end-users, and therefore imparts a degree of trust. The deceptive similarity can trick the user by embedding the essential parts of the target name, in a new string, or it can use some variant of the target name, such as replacing 'i' with 'l'.*” As an example, an email from `someone@example.net` (using a “one” character instead of the letter “l”) might easily be mistaken for the legitimate `someone@example.net` even if digitally signed by the fraudulent domain owner.

S/MIME by itself has no policy directives or feedback mechanisms. Automated policy enforcement could tighten the controls on acceptance or rejection of emails and provide feedback on failure mechanisms. A simple example would be to create a mailbox for a user that *only* accepts digitally signed email. All others (e.g. spear phishing messages) would be rejected. Another policy could be to enforce sender signing and encryption.

Sections 3 and 4 will describe methods to overcome these obstacles to make trusted email pervasive.

2.2. SPF, DKIM and DMARC

Because of the enormous growth in spam and phishing, various methods have been developed to limit their propagation. All of these methods use the DNS to publish and retrieve IETF standard records that dictate policy to an email server. Organizations such as the Anti-Phishing Working Group (APWG) and the Mobile, Messaging and Mail Anti-Abuse Working Group (M3AAWG) have encouraged their adoption. Although very useful in the context of spam, note that these do not constitute a full trust model for email. However they do complement S/MIME and DANE and would be incorporated in a comprehensive email solution. They are described here for completeness.

Sender Policy Framework (SPF) [17, 18] is a simple method to detect email spoofing by letting a sending domain identify and assert the authorized mail senders for a given domain. SPF removes guesswork as to the authenticity of a sending email server. This benefits receivers by allowing greater accuracy in quarantining and blocking.

DomainKeys Identified Mail (DKIM) [19] is a method to detect email spoofing by checking whether incoming mail from a domain has ben actually sent from that domain. Authorized sending email servers cryptographically sign all email headers (and email bodies) with the domain's private key. This signature allows the receiver to verify that email purported to come form a specific domain is authorized by the owner of the domain. It also allows verification as to whether headers or the message body was tampered with after it left the sending email server. The private key used to generate signatures is common to all email messages from that server. This means that DKIM does not offer true end-to-end digital signing, as the sending MTA generates the DKIM signature, not the original sender of the message. Verification is carried out at the receiving MTA using the domain's public key that is published in the DNS.

A problem with SPF and DKIM is the lack of feedback regarding its effectiveness. How many emails were blocked? Were mistakes made in setting policies or have all authorized senders been accounted for? Can a domain test the effectiveness of DKIM before fully turning it on?

Domain-based Message Authentication, Reporting, and Conformance (DMARC) [20] was defined to address these issues. DMARC was conceived to allow email senders to specify policy on how their mail

should be handled, the types of reports that receivers can send back and the frequency of reports. DMARC allows domain owners to know the extent to which unauthorized senders are using their domain.

2.3. Proprietary Systems

A number of commercial and open source products have been created to fill the void in email security. These can be appliances or cloud-based SaaS (Software as a Service). Systems include firewall products from FireEye, Cisco, SonicWall and others. SpamHaus, Sophos and Barracuda produce real-time query systems to determine if email is coming from a non-trusted source. AntiSpam protection and email security gateways are available from MXLogic (acquired by Intel), TrendMicro, FortiNet and others.

Proprietary email encryption products have been created due to the S/MIME limitations outlined earlier. Zix and ProofPoint are example products used by companies that need a fully functional email encryption solution. These are closed systems, however, and all parties have to use the same solution environment. Typically used in the financial sector, these proprietary solutions can be complex, and are neither universally available across diverse groups nor interoperable due to their walled-garden nature.

3. DANE

Supply Chain Management is a global process. Its diverse community of suppliers, customers and integrators typically use differing processes and systems. Interoperability of trusted email across this community is an absolute requirement. Proprietary solutions are inadequate due to their closed nature. A standards-based approach to trusted email, on the other hand, achieves universality and interoperability.

This leads to using standard S/MIME; it already exists and is available across all mail servers and clients. In fact, S/MIME can be and is used today, but the challenge in managing key distribution makes global scaling difficult. This limitation can be overcome, however, by means of the DANE protocol. DANE uses the global DNS infrastructure to overcome key distribution issues. It also solves problems in securing communication between mail exchange servers. Its use of the *existing* DNS infrastructure implies that solutions are readily deployable and affordable.

3.1. The DANE Mechanism

DNS-based Authentication of Named Entities (DANE, RFC6698) [7, 8] is a mechanism used to bind X.509 certificates into the DNS. The records are made cryptographically secure via the DNSSEC security extensions [21, 22, 23]. DANE can be used to store self-signed certificates, or to authorize specific X.509 certificates from a registered CA. It does this by publishing the X.509 certificate (or fingerprint thereof) in the appropriate specialized DANE resource record according to its usage: `TLSA` for certificates used to support TLS in applications, `OPENGPGKEY` or `SMIMEA` to support OpenPGP and S/MIME respectively.

One motivation for creating DANE was to solve issues with the existing X.509 Public Key Infrastructure (PKI). DANE, for example, addresses rejection of fraudulent certificates, permits simpler handling of certificate revocation, creates a mechanism for global publishing and retrieval of certificates, and allows the authorization of self-signed certificates.

DANE achieves these goals by using the *delegation property* of the DNS name space, meaning that only authorized domain owners can place records in their DNS domain. As an example, only the "example.com" corporation can place records in the `example.com` DNS name space. No one else can do so because they do not have access to the delegation. Delegation enables the creation of an authorization mechanism.

The first application of the DANE protocol was for the authentication of TLS certificates used by web servers. Consider a web site `www.example.com`. Assume that multiple certificates exist for that site, a real one and several fraudulent ones used by attackers for man-in-the-middle attacks (MITM). How can `www.example.com` protect itself? The solution is for the domain owner to insert a DANE `TLSA` record in the `www.example.com` DNS namespace to authorize only the genuine certificate. Web clients that retrieve certificates from a server can also retrieve the DANE record and match it against the certificate. If the DANE record exists and matches, the certificate is authorized and the connection is accepted. If the record does not match, the certificate is rejected and the connection is denied.

The DANE protocol is meant to be generic and multi-purpose. Application-specific use of DANE is defined in separate RFCs. Email usage is defined in two documents: RFC 7672 [24] defines `TLSA` records to secure the SMTP protocol for email servers, and an IETF draft document [9] defines `SMIMEA` records to secure end-user email certificates. We will explore each of these in turn.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3.2. DANE for MTA-MTA Security

A simplified email architecture is illustrated in figure 1. Email clients are programs such as Outlook, Apple Mail or Thunderbird that run in user devices (smartphones, tablets, laptops) to compose, send and retrieve email. Email is sent from these clients to Mail Transfer Agents (MTA) that store and forward the messages among themselves and finally to the recipient email client.

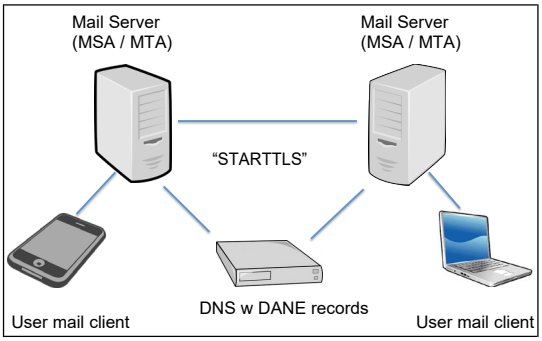


Figure 1: Simplified Email Architecture

Mail Transfer Agents will encrypt data sent from one MTA to another if TLS is available. This is a privacy measure for data-in-motion only. Once transferred, the data-at-rest is in plaintext.

Unfortunately, The original SMTP protocol did not accommodate TLS. To fix this, a new command, `STARTTLS`, was added to the protocol. `STARTTLS` modifies an existing insecure connection and upgrades it to a secure connection using SSL/TLS. The `STARTTLS` implementation, however, employs *opportunistic TLS*; that is, the receiving server can refuse the command and data communications between the two servers will continue in plaintext.

Opportunistic TLS creates vulnerability. An attacker can use a man-in-the-middle *downgrade* attack by simplify refusing the `STARTTLS` request. This allows eavesdropping and potential message modification by an attacker.

DANE eliminates this vulnerability as illustrated by the block diagram in figure 1. Before issuing a `STARTTLS`, the sending mail server will query the DNS for the DANE `TLSA` record associated with the receiving server. If a record exists, `STARTTLS` becomes *mandatory*. If a server refuses the `STARTTLS` request or if the certificate does not match the DANE `TLSA` record, communication between the servers will cease and the email server will wait to send the message at a later time. If a `TLSA` record does not exist, opportunistic TLS is still used. The absence or presence of a `TLSA` record permits incremental deployment of this DANE security mechanism.

DANE therefore achieves two goals for MTAs: it authenticates the receiver (certificate match), and enforces confidentiality via encryption between MTAs. Several email servers have already been modified to take advantage of this capability, including the popular open-source *Postfix* server.

3.2.1 Current Deployment of MTA-MTA Security Using DANE

The use of DANE for SMTP was specified in 2015 so deployment has been sparse as developers add the functionality to their implementations. There has been a sizable deployment within Germany and some experiences have been documented [25]. Using TLSA RRs to publish certificate information has been called out by the German Federal Office of Information Security as mandated for deployment as part of the "Email Made in Germany" initiative [26].

3.3. DANE for End-User Email Security

As mentioned, DANE for MTAs protects data-in-motion only. It does nothing for end-user authentication, digital signatures or data-at-rest encryption. For this we must use S/MIME. But the challenge has always been key management and distribution.

Assume employees in two organizations, `purple.com` and `green.com` need to communicate with each other using confidential and authenticated email. The employees have already obtained X.509 certificates. But how do personnel at either company obtain access to the public certificates of employees from the other company? There is no global public repository or "certificate phone book", where one can easily look up this information. As we explain below, however, DANE does provide just such a capability by publishing records in the global DNS.

Internet draft [9] extends DANE by defining the `SMIMEA` record. `SMIMEA` follows the same format as a `TLSA` record, but is used to store X.509 certificate data for individual users. The draft also defines a method to convert an email address, `john.doe@purple.com` into a domain name. The domain name uses a truncated SHA-256 hash of the user name to provide rudimentary privacy. The data stored in the `SMIMEA` record could be a complete X.509 certificate or a fingerprint. The DNS, secured by DNSSEC, is now a trusted repository or an authentication method for end user email certificates.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

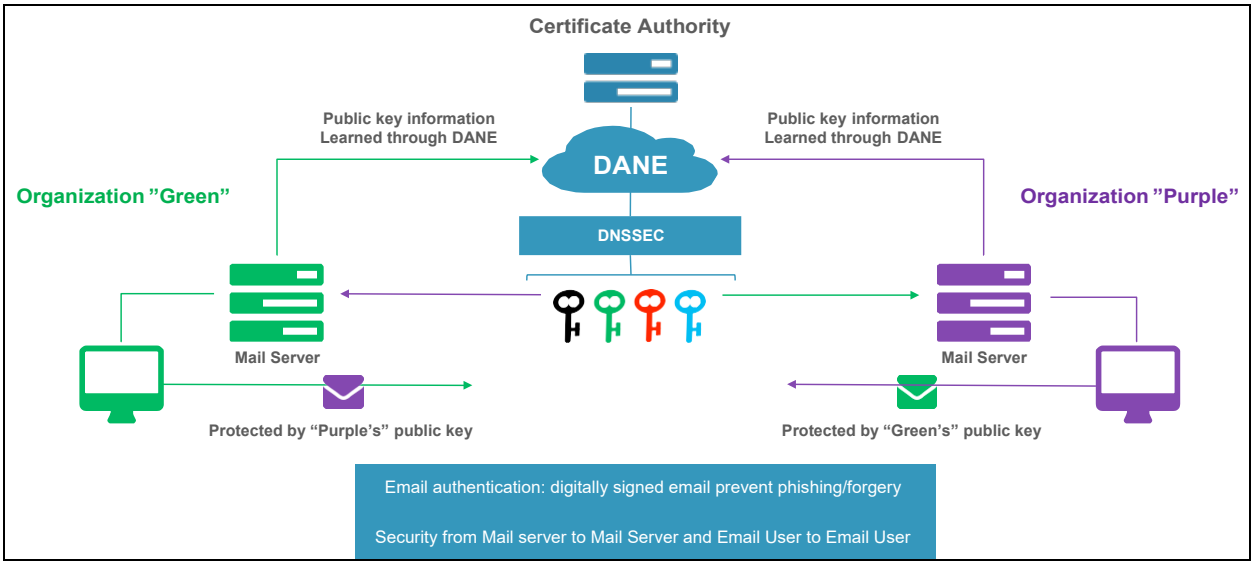


Figure 2: DANE system showing certificate retrieval for MTA and end-users (source: NIST [27])

The process for end-user security is illustrated in figure 2. A user at `green.com` digitally signs a message with her private key. This is done directly in the email client on her device. Next, in order to encrypt the message, a query to the DNS is made to retrieve the recipient's public key certificate. This certificate is cached by the user for future use and used to encrypt the message. Using the public key certificate ensures that only the recipient can decrypt the message. Performing the operation in the recipient's device ensures data-at-rest confidentiality. The signed and encrypted message is then transferred to the recipient through another encryption layer at the MTA to MTA level.

When the employee at `purple.com` receives the message, it is decrypted on his device via his private key. The user now needs to authenticate this message; did it really come from the sender at `green.com`, or is it a cleverly crafted spear phishing message? To confirm authenticity, the email program must check the veracity of the digital signature. This is done by performing a DNS lookup of the sender's public key certificate. The public key is used to decrypt the digital signature and perform a data integrity check. If the signature validates correctly, the message is authentic. It has not been altered in transit and the originator has been confirmed.

3.4. Policy: DMARC applied to DANE

As described in section 2.2, DMARC defines policy directives dictating the behavior of DKIM. It also provides a feedback mechanism to report on actual behavior. DANE can benefit from a similar mechanism. To that end, a draft proposing DMARC extensions for DANE [28] is currently a work-in-progress at the IETF. Sample policy directives include

- Receiver mail must be signed
- Receiver mail must be encrypted
- Sender mail must be signed
- Sender mail must be encrypted

Without these policies, users have to be attentive as to whether a received email has been digitally signed, typically indicated by an icon somewhere in the message. These indicators are easy to miss. Unsigned spear phishing messages without the icon could arrive unnoticed and potentially be acted upon.

To build strong protection, an organization could construct two inboxes for users. The "protected" inbox would enforce strict policy dictating all incoming email must be signed. The "unsafe" inbox accepts all mail, signed or unsigned. Official company business would be conducted within the protected mailbox. Other business could still be handled in the unsafe mailbox, but users now have the burden of checking for signatures.

Feedback mechanisms are currently being defined, but typically would report on various metrics such as failure counts, etc.

3.5. Objections and Alternatives to DANE

While this paper advocates the usage of DANE, there are several criticisms of the method. The Internet blog articles [29, 30] discuss its dependency on

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

DNSSEC and prompted many pro and con arguments. It should be noted that [30] limits its discussion to DANE for web site validation, as SMIMEA had not yet been introduced.

A counter-proposal for securing MTA-to-MTA communication has been proposed in a working draft at the IETF [31] for SMTP Strict Transport Security (STS). Like DANE, DKIM and DMARC, this protocol also publishes records in the DNS, however STS does not require DNSSEC. SMTP-STS is similar to STS for web servers, but modified for relevancy to SMTP. It works by having the receiving domain publish its security policy at a well-defined URL, which a sender accesses using HTTPS. Advantages are that it defines policies and feedback reporting and does not mandate the use of DNSSEC. Disadvantages are that it can be spoofed or DDoSed (Distributed Denial of Service) to make it appear that a policy is nonexistent. In addition, sending MTAs must now use HTTPS to insure that a secure channel exists. In contrast, DANE with DNSSEC has secure responses and proof of nonexistence built in.

STS is a trust mechanism for MTA to MTA only. It does nothing for client certificates used for end-to-end encryption and digital signing. SMIMEA remains as a viable key distribution method.

Research on the robustness, security, resilience and efficiency of DANE are only beginning at this time. This is a topic for future development. Current pilot programs are focusing on interoperability and core features.

3.6. Deployments & Government Programs

DANE can be deployed today and multiple organizations have already done so. The Internet Society Deploy360 Programme has created a website [32] listing some current deployments.

Of particular note is the trusted email showcase and testbed at NIST's National Cybersecurity Center of Excellence (NCCoE) [33]. The purpose of this NCCoE project is to demonstrate interoperability among commercially available DANE technologies from various suppliers. The use and setup of these technologies is being prepared to help government and private enterprise deploy DANE on their own.

The testbed has several environments contributed by Microsoft, Secure64, NLNetLabs, and ISC-Bind. Each environment contains DNSSEC servers, email servers, and email clients making use of DANE. Email can be exchanged between the environments to demonstrate interoperability in MTA-to-MTA security as well as end-user security with DANE S/MIME.

NIST has also published an excellent reference document to describe the principles and techniques currently available for secure email: *Trusted Email (Special Publication SP-800-177)* [27].

Other government involvement includes the drafting of proposed mandates that require DANE. The German government has published *BSI TR-03108 Secure E-mail Transport* [26], dated August 2015, requiring the use of DANE.

4. Workflow Automation

DANE removes the biggest limitation to using S/MIME on a global scale by creating a secure public repository of email certificates. The other limitations listed in section 2.1 still need to be addressed, as well as methods to make DANE easier to use. There are complexities in its use that begs for a more complete and automated solution.

As an example, an organization could use DANE as it exists today, however deployment would likely be limited to a small scale. This is because `TLSA` and `SMIMEA` records have to be manually generated and maintained. Mistakes are easy to make. Managing a trusted email environment will be difficult without proper tools and processes.

The objectives for managing a trusted email environment include the ability to automate DNS provisioning, integrate company workflows, simplify end-user activities, and manage company policy. To that end, the following items are being developed or already exist to assist DANE deployment and operations:

- Automated DNS zone file provisioner for TLSA, SMIMEA, SPF, DKIM and DMARC records.
- Automated DNSSEC signing appliances (e.g. Secure64, OpenDNSSEC) or DNSSEC enabled managed DNS services.
- Interfaces to Human Resource credential databases (e.g. Active Directory, etc.)
- Means to acquire or generate X.509 certificates either with an API to commercial CA accounts, an enterprise local CA, or tools to generate self-signed certificates.
- GUI objects and wizards to manage trust policies expressed as DMARC and DMARC/DANE records. Interfaces to legacy DMARC generation systems.
- Mail servers with milters for handling DANE and DMARC policy directives.
- Blacklist managers to block cousin domains.
- Integration with legacy email gateways and DMARC data collectors / report generators (e.g. Agari and others).
- Mobile Device Managers that provision personal devices with X.509 certificates and interface to

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

the DNS provisioner to exchange publishable trust data.

- Key escrow as an option for organizations whose policy requires that they have copies of end-user's private keys.
- Logging and Auditing with interfaces to SIEM systems.

A critical solution component is a DNS Provisioning System. The basic function of DNS record provisioning is to manage the workflow for creation and maintenance of DNS records and zone files. This task must be done with little or no manual intervention. Software API's link the provisioner to other components such as employee databases (e.g. *Active Directory*), mobile device managers (MDM), and Certificate Authority APIs. The provisioner could run locally or as a service in the cloud. The workflows to be managed are:

- *MTA management*: Automate discovery of mail servers and provisioning of TLSA, DKIM, DMARC and SPF records. Dynamically Maintain records due to external events such as server changes or certificate expiration or revocation. Interface with commercial systems for DMARC feedback (e.g. Agari). Software wizards and GUI objects would be needed to assist in policy definition for DANE and DKIM as well as interfaces to commercial systems for this function.
- *Employee Credential Management*:
  - *Initial setup*: Scrape the employee database to create S/MIMEA records. Company policy would dictate if private keys are owned by the corporation or by the individual. If the corporation owns the keys, certificates can be generated by a central system. If the individual owns the private keys, then only the individual's end user device should create key-pairs and the provisioner device API or MDM API will fetch this from user devices.
  - *Employee hire or termination*: This is best handled directly in the HR department through an API. Credentials would be established or revoked and the provisioning system would update its local database and DNS zone files on demand.
  - *Certificate renewal or revocation*: Manage the integration with HR databases, Certificate Authorities, and MDM to end-user devices; update records in the DNS.

Provisioning handles the supply side of certificate publication. The retrieval side is also in need of automation and simplification. Modern email clients such as *Outlook, Office 365, Apple Mail,* and *Thunderbird* have built-in encryption and digital signature verification using S/MIME. What they do not currently have is the ability to automatically retrieve DANE-formatted public keys from the DNS as mail is being composed and sent.

End-users should be able to use email with little or no change to existing email usage. This requires transparent integration of DANE into the end-user devices and easier methods to install user credentials.

It is expected that vendors of mobile and PC-based email clients will add this capability. In the temporary absence of such systems, simple standalone applications can fetch credentials from the DNS to store public certificates in the end-user device. Plugins to mailers such as *Thunderbird* have already been developed to make this step automatic.

4.1. Incremental Deployment

These suggestions for automation and usability may take some time to be fully realized. The authors wish to emphasize, however, that the core elements to build a DANE-enabled email system using manual steps is immediately available. Additional functionality can be implemented incrementally over time as new tools become available. A possible sequence of events is as follows:

1) Manual provisioning of DANE records: Early adopters are demonstrating the benefits of DANE, but manual implementation is impractical for the wider audience. Nevertheless, scripts are currently available for constructing TLSA and SMIMEA records and these can be installed in an organization's authoritative DNS servers immediately.

2) Manual retrieval of certificates: some standalone apps and email client plugins are available and more are under development. These programs access email contact lists from user devices to fetch email credentials and insert them into the device's keystore. These tools require manual intervention by the user rather than the more desirable goal of transparent fetching of credentials within an email client.

3) Mail Filters (Milters) to automatically fetch encryption certificates at the mail server. SMILLA [34] is an existing example milter that encrypts mail at the server instead of at the sending email client. This provides a "90% solution"; that is, it provides data-at-rest protection at the end user device with messages

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

uniquely encrypted for the recipient, but does not perform encryption at the first mile between the sending device and the mail server. The solution is still useful, however, since this first mile communication is typically encrypted with TLS.

4) Future automated provisioning of SMIMEA records: organizations will be able to publish their employee certificates more easily, but the recipients of email will still have no client applications (e.g. Thunderbird, Exchange, Apple Mail) to automatically retrieve these certificates.

5) Future fully transparent email client integration: No manual intervention required by the end-user to retrieve public keys.

6) Future security policy Enforcement on servers and end-user devices: creates the possibility of inboxes that only accept signed and/or encrypted email.

5. Conclusion

The authors have demonstrated the need for trusted email in supply chain management. Spear phishing, forgery, and other attacks can result in data breaches, industrial and government espionage, installation of malware, and financial theft. DANE email extensions are then posited as a solid foundation for global trusted email. DANE creates a secure repository for publishing and retrieving email credentials and policy directives on a globally scalable basis.

Research on the efficiency, security and robustness of DANE email, as well as in-depth comparisons to other technologies is only in the starting phases. Current pilots are focusing on interoperability and core functionality. This is a topic for future development.

Despite its promising capabilities, however, the basic DANE protocol is nothing more than an enabling technology at this time. A complete trusted email solution will require the development of an ecosystem of automated tools, procedures and the incorporation of new DANE features into existing popular email client programs.

To this end, supplementary automation tools to manage the workflow of DANE are proposed. The tools discussed are for both sides of the equation: the automated publishing of email certificates as well as email client features and plugins to simplify the retrieval of certificates and make user interaction with trusted email as transparent as possible. Some components are under development; some already exist.

Finally, the authors encourage supply chain managers to reduce risk by protecting their email with basic DANE technology as soon as possible. Implementing DANE trusted email manually with the existing infrastructure components is definitely possible using published scripts and tools. The basic functionality can then be expanded into a more robust, automated, and easier to use solution for a more general audience as additional tools become available.

6. References

[1] McCaney, K. “To hackers, government users are phish in a barrel”. GCN, March 19, 2012. https://gcn.com/articles/2012/03/19/phishing-goverment-cyber-attacks-us-cert.aspx.

[2] OMB. Fiscal year 2011 report to congress on the implementation of the federal information security management act of 2002, 2011. https://www.whitehouse.gov/sites/default/files/omb/assets/egov docs/fy11 fisma.pdf.

[3] Cisco. Email Attacks: This Time It’s Personal. June 2011. http://www.cisco.com/c/dam/en/us/products/collateral/security/email- security-appliance/targeted attacks.pdf.

[4] TrendLabs APT Research Team. Spear phishing email:Most favored apt attack bait, 2012. http://www.trendmicro.com/cloud- content/us/pdfs/security-intelligence/white-papers/wp-spear phishing- email-most-favored-apt-attack-bait.pdf.

[5] Verizon. Data Breach Investigation Report, 2016. http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/

[6] M. Novinson, “Arrow was the target: Criminals impersonate executive, transfer money to outside bank”. CRN, February 4, 2016. http://www.crn.com/news/security/300079601/arrow-was-the-target- criminals-impersonate-executive-transfer-money-to-outside-bank.htm.

[7] J. Schlyter and P. Hoffman. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, August 2012. https://tools.ietf.org/html/rfc6698

[8] Shumon Huque, Dan James, and Viktor Dukhovni. TLS Client Authentication via DANE TLSA records. Internet-Draft draft-huque-dane-client-cert-02, Internet Engineering Task Force, January 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-huque-dane-client-cert/

[9] J. Schlyter and P. Hoffman. Using Secure DNS to Associate Certificates with Domain Names For S/MIME.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Internet-Draft draft-ietf- dane-smime-10, Internet Engineering Task Force, February 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-ietf-dane-smime/

[10] J. Klensin. Simple Mail Transfer Protocol. RFC 5321, October 2008. https://tools.ietf.org/html/rfc5321

[11] N. Freed and Dr. N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045, November 1996. https://tools.ietf.org/html/rfc2045

[12] N. and Dr. N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC 2046, November 1996. https://tools.ietf.org/html/rfc2046

[13] K. Moore. MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text. RFC 2047, November 1996. https://tools.ietf.org/html/rfc2047

[14] S. Turner and B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling. RFC 5750, January 2010. https://tools.ietf.org/html/rfc5750

[15] S. Turner and B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. RFC 5751, January 2010. https://tools.ietf.org/html/rfc5751

[16] T. Roessler, M. Elkins, R. Levien, and D. Del Torto. MIME Security with OpenPGP. RFC 3156, August 2001. https://tools.ietf.org/html/rfc3156

[17] W. Schlitt and M. Wong. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408, April 2006. https://tools.ietf.org/html/rfc4408

[18] M. Kucherawy. Resolution of the Sender Policy Framework (SPF) and Sender ID Experiments. RFC 6686, July 2012. https://tools.ietf.org/html/rfc6686

[19] M. Kucherawy, D. Crocker, and T. Hansen. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376, September 2012. https://tools.ietf.org/html/rfc6376

[20] M. Kucherawy and E. Zwicky. Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489, March 2015. https://tools.ietf.org/html/rfc7489

[21] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends. DNS Security Introduction and Requirements. RFC 4033, March 2005. https://tools.ietf.org/html/rfc4033

[22] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends.. Resource Records for the DNS Security Extensions. RFC 4034, March 2005. https://tools.ietf.org/html/rfc4034

[23] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends.. Protocol Modifications for the DNS Security Extensions. RFC 4035, March 2005. https://tools.ietf.org/html/rfc64035

[24] V. Dukhovni and W. Hardaker. SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS). RFC 7672, October 2015. https://tools.ietf.org/html/rfc7672

[25] P. Koetter Sys4.de presentation at the 34th M3AAWG meeting “One year of DANE: Tales and Lessons Learned” June 2015 https://sys4.de/download/dane-maawg.pdf

[26] Federal Office of Information Security. BSI TR-03108-1: Secure E-Mail Transport (English Version). March 22, 2016. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03108/TR03108-1.pdf? blob=publicationFile&v=3

[27] R. Chandramouli, S. Garfinkel, S. Nightingale, and S. Rose. Trusted Email. SECOND DRAFT NIST Special Publication 800-177, March 2016. http://csrc.nist.gov/publications/PubsSPs.html

[28] E. Osterweil and G. Wiley. DMARC Extensions for DANE. Internet-Draft draft-osterweil-dmarc-dane-names-00, Internet Engineering Task Force, January 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-osterweil-dmarc-dane-names/

[29] T. Ptaceck and E. Ptacek. Against DNSSEC. Blog article, January 15, 2015. http://sockpuppet.org/blog/2015/01/15/against-dnssec/

[30] A. Langley. Why not DANE in Browsers. Blog article, January 17, 2015. https://www.imperialviolet.org/2015/01/17/notdane.html.

[31] D. Margolis, M. Risher, N. Lidzborkski, W. Chuang, B. Long, B. Ramakrishnan, A. Brotman, J. Jones, F. Martin, K. Umbach, and M. Laber. SMTP MTA Strict Transport Security. Internet-Draft draft-ietf-uta-mta-sts-00, Internet Engineering Task Force, May 2016. Work in Progress. https://datatracker.ietf.org/doc/draft-ietf-uta-mta-sts/

[32] ISOC Deploy360 Programme, DANE Test Sites. http://www.internetsociety.org/deploy360/resources/dane-test-sites/.

[33] W. Barker. Domain Name Based Security for Electronic Email, March 2016. https://nccoe.nist.gov/sites/default/files/library/project-descriptions/dns- secure-email-project-description-final.pdf.

[34] P.B. Koetter. Smilla – SMIMEA aware Milter. IETF DANE thread with github repository, July 2, 2015. https://www.ietf.org/mailarchive/web/dane/current/msg07865.html

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Key Recovery Attack on the Cubic ABC Simple Matrix Multivariate Encryption Scheme

Dustin Moody[1], Ray Perlner[1], and Daniel Smith-Tone[1,2]

[1]National Institute of Standards and Technology,
Gaithersburg, Maryland, USA
[2]Department of Mathematics, University of Louisville,
Louisville, Kentucky, USA

dustin.moody@nist.gov, ray.perlner@nist.gov, daniel.smith@nist.gov

Abstract. In the last few years multivariate public key cryptography has experienced an infusion of new ideas for encryption. Among these new strategies is the ABC Simple Matrix family of encryption schemes which utilize the structure of a large matrix algebra to construct effectively invertible systems of nonlinear equations hidden by an isomorphism of polynomials. The cubic version of the ABC Simple Matrix Encryption was developed with provable security in mind and was published including a heuristic security argument claiming that an attack on the scheme should be at least as difficult as solving a random system of quadratic equations over a finite field.
In this work, we prove that these claims are erroneous. We present a complete key recovery attack breaking full sized instances of the scheme. Interestingly, the same attack applies to the quadratic version of ABC, but is far less efficient; thus, the enhanced security scheme is less secure than the original.

Key words: multivariate public key cryptography, differential invariant, MinRank, encryption

1 Introduction

Classical public key cryptography is mainly based on arithmetic constructions on Abelian groups. Since the discovery by Peter Shor in the 1990s of efficient algorithms for factoring and computing discrete logarithms with quantum computers, see [1], there has been a growing interest in the international community in the task of constructing algorithms resistant to cryptanalysis with quantum computers. Indeed, in light of the announcement [2] by the National Institute of Standards and Technology (NIST) of an imminent call for proposals for post-quantum standards, the challenge of migrating from the homogeneous heritage of public key cryptography to a more diverse collection of tools has become mainstream.

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

One possible candidate for practical, efficient, and nonconforming solutions to some of the most consequential public key applications is Multivariate Public Key Cryptography(MPKC). Multivariate schemes are attractive in certain applications because of the maleability of the schemes. Different modifications of similar ideas can make a scheme more suited to lightweight architectures, enhance security, or parametrize various aspects of performance.

In addition, MPKC is one among a few serious candidates to have risen to prominence as post-quantum options. The fundamental problem of solving a system of quadratic equations is known to be NP-hard, and so in the worst case, solving a system of generic quadratic equations is unfeasible for a classical computer; neither is there any indication that the task is easier in the quantum computing paradigm.

MPKC has experienced a fair amount of success in the realm of digital signatures. Some trustworthy schemes that have survived for almost two decades include UOV [3], HFE- [4], and HFEv- [5]. Moreover, some of these schemes have optimizations which have strong theoretical support or have stood unbroken in the literature for some time. Specifically, UOV has a cyclic variant [6] which reduces the key size dramatically, and Gui, a new HFEv- scheme, see [7], has parameters far more appealing than QUARTZ due to greater confidence in the complexity of algebraically solving the underlying system of equations [8].

The situation with multivariate public key encryption is quite different, however. Many attempts at multivariate encryption, see [9, 10] for example, have been shown to be weak based on rank or differential weaknesses. Recently, a few interesting attempts to achieve multivariate encryption have surfaced. ZHFE, see [11], and the ABC Simple Matrix Scheme, see [12], both use fundamentally new structures for the derivation of an encryption system. While it was shown that the best attack known on the Simple Matrix structure, see [13] — which relies on the differential invariant structure of the central map — supports the claimed security level of the scheme, a subset of the original authors proposed a cubic version of the scheme, [14], as a step towards provable security.

In this article, we present a key recovery attack on a full scale version of the Cubic Simple Matrix encryption scheme, having a complexity on the order of q^{s+2} for characteristic $p > 3$, q^{s+3} for characteristic 3 and q^{2s+6} for characteristic 2. Here s is the dimension of the matrices in the scheme, and q is the cardinality of the finite field used. This technique is an extension and augmentation of the technique of [13], and, similarly, exploits a differential invariant property of the core map to perform a key recovery attack. We can show that the attack uses a property which uniquely distinguishes the isomorphism class of the central map from that of a random collection of formulae.

Specifically, our attack breaks CubicABC($q = 127$,$s = 7$), designed for 80-bit security, in approximately 2^{76} operations (or around 2^{80} if one pessimistically uses $\omega = 3$ as the linear algebra constant). More convincingly, our attack completely breaks CubicABC($q = 127$,$s = 8$), designed for 100-bit security, in approximately 2^{84} operations (or 2^{88} for $\omega = 3$). Furthermore, the attack is fully parallelizable and requires very little memory; hence, the differential invariant

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

attack is far more efficient than algebraic attacks, the basis for the original security estimation. Thus, the security claims in [14] are clearly unfounded; in fact, the cubic version of the scheme, whose security was claimed to be closely related to an NP-complete problem, is actually less secure than the quadratic case.

The paper is organized as follows. In the next section, we present the structure of the Cubic ABC Simple Matrix encryption scheme. In the following section, we recall differential invariants and present a natural extension of this notion to the case of cubic polynomials. The differential invariant structure of the ABC scheme is derived in the subsequent section and the effect of this structure on minrank calculations is determined. We next calculate the complexity of the full attack including the linear algebra steps required to extend the distinguisher into a key recovery mechanism. Finally, we review these results and discuss the surprising relationship between the practical security of the Cubic ABC scheme and its quadratic counterpart.

2 The Cubic ABC Matrix Encryption Scheme

In [14], the Cubic ABC Matrix encryption scheme is proposed. The motivation behind the scheme is to use a large matrix algebra over a finite field to construct an easily invertible cubic map. The construction uses matrix multiplication to combine random quadratic formulae and random linear formula into cubic formulae in a way that allows a user with knowledge of the structure of the matrix algebra and polynomial isomorphism used to compose the scheme to invert the map.

Let $k = \mathbb{F}_q$ be a finite field. Linear forms and variables over k will be denoted with lower case letters. Vectors of any dimension over k will be denoted with bold font, $\mathbf{v}$. Fix $s \in \mathbb{N}$ and set $n = s^2$ and $m = 2s^2$. An element of $M_s(k)$, $M_n(k)$ or $M_m(k)$, or the linear transformations they represent, will be denoted by upper case letters, such as M. When the entries of the matrix are being considered functions of a variable, the matrix will be denoted $M(\mathbf{x})$. Let ϕ represent the vector space isomorphism from $M_{s\times 2s}(k)$ to k^{2s^2} sending a matrix to the column vector consisting of the concatenation of its rows. The output of this map, being a vector, will be written with bold font; however, to indicate the relationship to its matrix preimage, it will be denoted with an upper case letter, such as $\mathbf{M}$.

The scheme utilizes an isomorphism of polynomials to hide the internal structure. Let $\mathbf{x} = \left[x_1, x_2, \ldots, x_n\right]^\top \in k^n$ denote plaintext while $\mathbf{y} = \left[y_1, \ldots, y_m\right] \in k^m$ denotes ciphertext. Fix two invertible linear transformations $T \in M_m(k)$ and $U \in M_n(k)$. (One may use affine transformations, but there is no security or performance benefit in doing so.) Denote the input and output of the central map by $\mathbf{u} = U\mathbf{x}$ and $\mathbf{v} = T^{-1}(\mathbf{y})$.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

The construction of the central map is as follows. Define three $s \times s$ matrices A, B, and C in the following way:

$$A = \begin{bmatrix} p_1 & p_2 & \cdots & p_s \\ p_{s+1} & p_{s+2} & \cdots & p_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ p_{s^2-s+1} & p_{s^2-s+2} & \cdots & p_{s^2} \end{bmatrix}, B = \begin{bmatrix} b_1 & b_2 & \cdots & b_s \\ b_{s+1} & b_{s+2} & \cdots & b_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ b_{s^2-s+1} & b_{s^2-s+2} & \cdots & b_{s^2} \end{bmatrix},$$

and

$$C = \begin{bmatrix} c_1 & c_2 & \cdots & c_s \\ c_{s+1} & c_{s+2} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{s^2-s+1} & c_{s^2-s+2} & \cdots & c_{s^2} \end{bmatrix}.$$

Here the p_i are quadratic forms on $\mathbf{u}$ chosen independently and uniformly at random from among all quadratic forms and the b_i and c_i are linear forms on $\mathbf{u}$ chosen independently and uniformly at random from among all linear forms.

We define two $s \times s$ matrices $E_1 = AB$ and $E_2 = AC$. Since A is quadratic and B and C are linear in u_i, E_1 and E_2 are cubic in the u_i. The central map $\mathcal{E}$ is defined by

$$\mathcal{E} = \phi \circ (E_1 || E_2).$$

Thus $\mathcal{E}$ is an m dimensional vector of cubic forms in $\mathbf{u}$. Finally, the public key is given by $\mathcal{F} = T \circ \mathcal{E} \circ U$.

Encryption with this system is standard: given a plaintext $(x_1, \ldots, x_n)$, compute $(y_1, \ldots, y_m) = \mathcal{F}(x_1, \ldots, x_n)$. Decryption is somewhat more complicated.

To decrypt, one inverts each of the private maps in turn: apply T^{-1}, invert $\mathcal{E}$, and apply U^{-1}. To "invert" $\mathcal{E}$, one assumes that $A(\mathbf{u})$ is invertible, and forms a matrix

$$A^{-1}(\mathbf{u}) = \begin{bmatrix} w_1 & w_2 & \cdots & w_s \\ w_{s+1} & w_{s+2} & \cdots & w_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ w_{s^2-s+1} & w_{s^2-s+2} & \cdots & w_{s^2} \end{bmatrix},$$

where the w_i are indeterminants. Then using the relations $A^{-1}(\mathbf{u})E_1(\mathbf{u}) = B(\mathbf{u})$ and $A^{-1}(\mathbf{u})E_2(\mathbf{u}) = C(\mathbf{u})$, we have $m = 2s^2$ linear equations in $2n = 2s^2$ unknowns w_i and u_i. Using, for example, Gaussian elimination one can eliminate all of the variables w_i and most of the u_i. The resulting relations can be substituted back into $E_1(\mathbf{u})$ and $E_2(\mathbf{u})$ to obtain a large system of equations in very few variables which can be solved efficiently in a variety of ways.

3 Subspace Differential Invariants for Cubic Maps

Let $f : k^n \to k^m$ be an arbitrary fixed function on k^n. Consider the discrete differential $Df(\mathbf{a}, \mathbf{x}) = f(\mathbf{a} + \mathbf{x}) - f(\mathbf{a}) - f(\mathbf{x}) + f(\mathbf{0})$.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

If f is quadratic, we can express the differential as an n-tuple of bilinear differential coordinate forms in the following way: $[Df(\mathbf{a},\mathbf{x})]_i = \mathbf{a}^\top Df_i\mathbf{x}$, where Df_i is a symmetric matrix representation of the action on the ith coordinate of the bilinear differential. If the function f is cubic $Df(\mathbf{a},\mathbf{x})$ is a symmetric bi-quadratic function. By the symmetry, it is well defined to compute a second differential $D^2f(\mathbf{a},\mathbf{b},\mathbf{x})$ by computing the discrete differential of Df with respect to either $\mathbf{a}$ or $\mathbf{x}$. In this case, we may consider the second differential as an n-tuple of trilinear differential coordinate forms by letting D^2f_i be the symmetric 3-tensor representing the action on the ith coordinate of the trilinear differential.

In [13], the following definition of a subspace differential invariant was provided:

Definition 1 *A* subspace differential invariant *of a quadratic map $f : k^n \to k^m$ with respect to a subspace $X \subseteq k^m$ is a subspace $V \subseteq k^n$ with the property that there exists a $W \subseteq k^n$ of dimension at most $dim(V)$ such that simultaneously $AV \subseteq W$ for all $A = \sum_{i=1}^m x_i Df_i$ where $(x_1,\ldots,x_m) \in X$, i.e. $A \in Span_X(Df_i)$.*

This definition captures the idea of a subspace of the span of the public polynomials acting linearly on a subspace of the plaintext space in the same way. Such behavior is strange for quadratic maps in general. Furthermore, as shown in [13], this behavior is computable regardless of the rank of the maps involved.

A natural generalization of this definition is the following:

Definition 2 *A* subspace differential invariant *of a cubic map $f : k^n \to k^m$ with respect to a subspace $X \subseteq k^m$ is a pair of subspaces $(V_1, V_2) \subseteq (k^n)^2$ for which there exists a subspace $W \subseteq k^n$ with $dim(W) \leq mindim(V_i)$ such that for all $A = \sum_{i=1}^m x_i D^2 f_i$ where $(x_1,\ldots,x_m) \in X$, for all $\mathbf{a} \in V_2$, for all $\mathbf{b} \in V_2$ and for all $\mathbf{x} \in W^\perp$ we have that $A(\mathbf{a},\mathbf{b},\mathbf{x}) = 0$.*

This definition captures the notion of a subspace of the span of the public cubic polynomials acting quadratically on a subspace of the plaintext space in the same way. Such behavior is strange for cubic maps in general.

4 The Differential Invariant Structure of the Cubic ABC scheme

4.1 Column Band Spaces

Each component of the central $\mathcal{E}(\mathbf{u}) = E_1(\mathbf{u})||E_2(\mathbf{u})$ map may be written as:

$$\mathcal{E}_{(i-1)s+j} = \sum_{l=1}^{s} p_{(i-1)s+l} b_{(l-1)s+j}, \tag{1}$$

for the E_1 equations, and likewise, for the E_2 equations:

$$\mathcal{E}_{s^2+(i-1)s+j} = \sum_{l=1}^{s} p_{(i-1)s+l} c_{(l-1)s+j} \tag{2}$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

where i and j run from 1 to s.

Consider the s sets of s polynomials that form the columns of E_1, i.e. for each $j \in \{1, \ldots, s\}$ consider $(\mathcal{E}_j, \mathcal{E}_{s+j}, \ldots, \mathcal{E}_{s^2-s+j})$. With high probability, the linear forms $b_j, b_{s+j}, \ldots, b_{s^2-s+j}$ are linearly independent, and if so the polynomials may be re-expressed, using a linear change of variables to $(u'_1, \ldots u'_{s^2})$ where $u'_i = b_{(i-1)s+j}$ for $i = 1, \ldots, s$. After the change of variables, the only cubic monomials contained in $(\mathcal{E}_j, \mathcal{E}_{s+j}, \ldots, \mathcal{E}_{s^2-s+j})$ will be those containing at least one factor of $u'_1, \ldots, u'_s$. We can make a similar change of variables to reveal structure in the s sets of s polynomials that form the columns of E_2: Setting $u'_i = c_{(i-1)s+j}$ for $i = 1, \ldots, s$ and a fixed j, the only cubic monomials contained in $(\mathcal{E}_{s^2+j}, \mathcal{E}_{s^2+s+j}, \ldots, \mathcal{E}_{2s^2-s+j})$ will be those containing at least one factor of $u'_1, \ldots, u'_s$.

More generally, we can make a similar change of variables to reveal structure in any of a large family of s dimensional subspaces of the span of the component polynomials of E_1 and E_2, which we will call column band spaces in analogy to the band spaces used to analyze the quadratic ABC cryptosystem in [13]. Each family is defined by a fixed linear combination, (β, γ), of the columns of E_1 and E_2:

Definition 3 *The column band space defined by the* $2s$*-dimensional linear form* (β, γ) *is the space of cubic maps,* $\mathcal{B}_{\beta,\gamma}$ *, given by:*

$$\mathcal{B}_{\beta,\gamma} = \mathit{Span}(\mathcal{E}_{\beta,\gamma,1}, \ldots, \mathcal{E}_{\beta,\gamma,s})$$

where

$$\mathcal{E}_{\beta,\gamma,i} = \sum_{j=1}^{s} (\beta_j \mathcal{E}_{(i-1)s+j} + \gamma_j \mathcal{E}_{s^2+(i-1)s+j})$$

$$= \sum_{l=1}^{s} \left(p_{(i-1)s+l} \sum_{j=1}^{s} \left(\beta_j b_{(l-1)s+j} + \gamma_j c_{(l-1)s+j} \right) \right)$$

Theorem 1 *There is a pair of subspaces* $(V_1, V_2) \in (k^n)^2$ *which is a subspace differential invariant with respect to* $\mathcal{B}_{\beta,\gamma}$ *for all* (β, γ)*. Moreover, there exists an* $\mathbf{x}_1 \in k^n$ *such that* $rank(D^2\mathcal{E}(\mathbf{x}_1)) \leq 2s$ *for all* $\mathcal{E} \in \mathcal{B}_{\beta,\gamma}$*.*

Proof. Note that under a change of variables $(x_1, \ldots, x_{s^2}) \xrightarrow{M} (u'_1, \ldots u'_{s^2})$, where $u'_i = \sum_{j=1}^{s} \left(\beta_j b_{(i-1)s+j} + \gamma_j c_{(i-1)s+j} \right)$ for $i = 1, \ldots, s$, the only cubic monomials contained in the elements of $\mathcal{B}_{\beta,\gamma}$ will be those containing at least one factor of $u'_1, \ldots, u'_s$. In such a basis, the second differential of any map in $\mathcal{B}_{\beta,\gamma}$, and thus the second differential of $\mathcal{E}$ can be visualized as a 3-tensor with a special block form, see Figure 1.

Let V be the $(s^2 - s)$-dimensional preimage $M^{-1}(\mathrm{Span}(u'_1, \ldots, u'_s)^\perp)$. This 3-tensor $D^2\mathcal{E}$ may be thought of as a bilnear map which takes two vectors $\mathbf{x}_1, \mathbf{x}_2 \in V$, i.e. of the form:

$$(0, \ldots, 0, u'_{s+1}(\mathbf{x}_k), \ldots, u'_{s^2}(\mathbf{x}_k))^\top$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

to a covector of the form:

$$(y(u'_1), \ldots, y(u'_s), 0, \ldots, 0).$$

Thus, in this basis $D^2\mathcal{E}(\mathbf{x}_1)$ is a symmetric matrix which is zero on $V \times V$. Therefore, $rank(D^2\mathcal{E}(\mathbf{x})) \leq 2s$. One checks that (V, V) is a subspace differential with respect to $\mathcal{B}_{\beta,\gamma}$ with $W := V^{\perp}$, since $dim(W) = s < s^2 - s = dim(V)$.

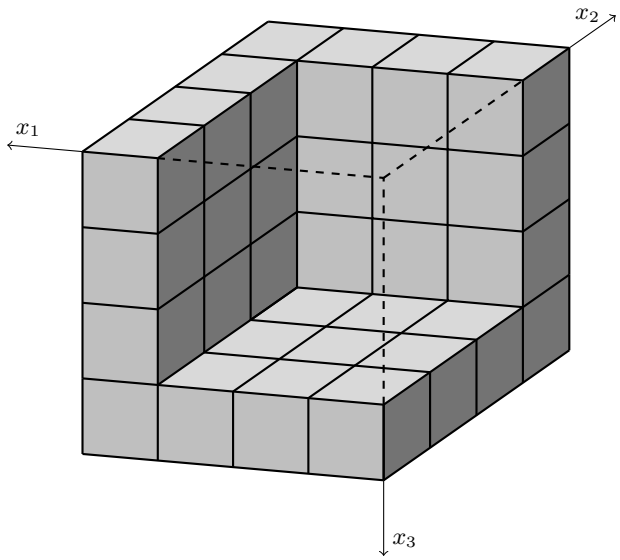


Fig. 1. 3-tensor structure of the second differential of a band space map. Solid regions correspond to nonzero coefficients. Transparent regions correspond to zero coefficients.

We will define the term "band-kernel" to describe the space of vectors of the same form as x_1 and x_2 in the proof above, i.e.:

Definition 4 *The band kernel of $\mathcal{B}_{\beta,\gamma}$, denoted $\mathcal{BK}_{\beta,\gamma}$, is the space of vectors, x, such that*

$$u'_i = \sum_{j=1}^{s} \left(\beta_j b_{(i-1)s+j}(x) + \gamma_j c_{(i-1)s+j}(x)\right) = 0$$

for $i = 1, \ldots, s$.

5 A Variant of MinRank Exploiting the Column Band Space Structure

A minrank-like attack may be used to locate the column band-space maps defined in the previous section. In this case, the attack proceeds by selecting s^2-dimensional vectors $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$, setting

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

$$\begin{aligned}\sum_{i=1}^{2s^2} t_i D^2\mathcal{E}_i(\mathbf{x}_1,\mathbf{x}_2) = 0\\ \sum_{i=1}^{2s^2} t_i D^2\mathcal{E}_i(\mathbf{x}_3,\mathbf{x}_4) = 0,\end{aligned} \tag{3}$$

and solving for the t_i. The attack succeeds when $\sum_{i=1}^{2s^2} t_i\mathcal{E}_i \in \mathcal{B}_{\beta,\gamma}$ and $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ are all within the corresponding band kernel. If these conditions are met, then the rank of the 2-tensor $\sum_{i=1}^{2s^2} t_i D^2\mathcal{E}_i(\mathbf{x}_k)$ for $k = 1, 2, 3, 4$ will be at most $2s$, and this will be easily detectable.

The attack complexity will be significantly reduced if several of the $\mathbf{x}_k$ are set equal to one another. In odd characteristic fields, we can reduce the number of independently chosen vectors to 2, (for example, by setting $\mathbf{x}_1 = \mathbf{x}_2$ and $\mathbf{x}_3 = \mathbf{x}_4$.) In characteristic 2, however, the antisymmetry of the 2nd differential prevents the equation $\sum_{i=1}^{2s^2} t_i D^2\mathcal{E}_i(\mathbf{x}_1,\mathbf{x}_1) = 0$ from imposing a nontrivial constraint on the t_i. Even in characteristic 2, though, the number of independently chosen vectors can be reduced to 3 (e.g. by setting $\mathbf{x}_1 = \mathbf{x}_4$.)

Theorem 2 *The probability that 2 randomly chosen vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, are both in the band kernel of some band-space $\mathcal{B}_{\beta,\gamma}$ is approximately $\frac{1}{q-1}$; The probability that 3 randomly chosen vectors, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, are all in the band kernel of some band-space $\mathcal{B}_{\beta,\gamma}$ is approximately $\frac{1}{(q-1)q^s}$.*

Proof. The condition that the $\mathbf{x}_k$ are all contained within a band kernel is that there be a nontrivial linear combination of the columns of the following matrix equal to zero (i.e. that the matrix has nonzero column corank):

$$\left[\begin{array}{cccc|cccc} b_1(\mathbf{x}_1) & b_2(\mathbf{x}_1) & \dots & b_s(\mathbf{x}_1) & c_1(\mathbf{x}_1) & c_2(\mathbf{x}_1) & \dots & c_s(\mathbf{x}_1)\\ b_{s+1}(\mathbf{x}_1) & b_{s+2}(\mathbf{x}_1) & \dots & b_{2s}(\mathbf{x}_1) & c_{s+1}(\mathbf{x}_1) & c_{s+2}(\mathbf{x}_1) & \dots & c_{2s}(\mathbf{x}_1)\\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots\\ b_{s^2-s+1}(\mathbf{x}_1) & b_{s^2-s+2}(\mathbf{x}_1) & \dots & b_{s^2}(\mathbf{x}_1) & c_{s^2-s+1}(\mathbf{x}_1) & c_{s^2-s+2}(\mathbf{x}_1) & \dots & c_{s^2}(\mathbf{x}_1)\\ \hline b_1(\mathbf{x}_2) & b_2(\mathbf{x}_2) & \dots & b_s(\mathbf{x}_2) & c_1(\mathbf{x}_2) & c_2(\mathbf{x}_2) & \dots & c_s(\mathbf{x}_2)\\ b_{s+1}(\mathbf{x}_2) & b_{s+2}(\mathbf{x}_2) & \dots & b_{2s}(\mathbf{x}_2) & c_{s+1}(\mathbf{x}_2) & c_{s+2}(\mathbf{x}_2) & \dots & c_{2s}(\mathbf{x}_2)\\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots\\ b_{s^2-s+1}(\mathbf{x}_2) & b_{s^2-s+2}(\mathbf{x}_2) & \dots & b_{s^2}(\mathbf{x}_2) & c_{s^2-s+1}(\mathbf{x}_2) & c_{s^2-s+2}(\mathbf{x}_2) & \dots & c_{s^2}(\mathbf{x}_2)\\ \hline \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \end{array}\right] \tag{4}$$

In the case with 2 randomly chosen vectors, the matrix is a uniformly random $2s \times 2s$ matrix, which has nonzero column corank with probability approximately $\frac{1}{q-1}$. In the case with 3 randomly chosen vectors, the matrix is a uniformly random $3s \times 2s$ matrix, which has nonzero column corank with probability approximately $\frac{1}{(q-1)q^s}$.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

Theorem 3 *If* $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, *and* $\mathbf{x}_4$ *are chosen in such a way that all four vectors are in the band kernel of a column band space* $\mathcal{B}_{\beta,\gamma}$ *and also that the symmetric tensor products* $\mathbf{x}_1 \odot \mathbf{x}_2$ *and* $\mathbf{x}_3 \odot \mathbf{x}_4$ *are linearly independent from one another and statistically independent from the private quadratic forms,* $p_{(i-1)s+j}$ *in the matrix A, then the tensor products* $\mathbf{x}_1 \otimes \mathbf{x}_2$ *and* $\mathbf{x}_3 \otimes \mathbf{x}_4$ *are both in the kernel of some column band-space differential* $D^2\mathcal{E} = \sum_{\mathcal{E}_{\beta,\gamma,i} \in \mathcal{B}_{\beta,\gamma}} \tau_i D^2\mathcal{E}_{\beta,\gamma,i}$ *with probability approximately* $\frac{1}{(q-1)q^s}$.

Proof. A $D\mathcal{E}$ meeting the above condition exists iff there is a nontrivial solution to the following system of equations

$$\begin{aligned} \sum_{\mathcal{E}_{\beta,\gamma,i} \in \mathcal{B}_{\beta,\gamma}} \tau_i D^2\mathcal{E}_{\beta,\gamma,i}(\mathbf{x}_1, \mathbf{x}_2) = 0, \\ \sum_{\mathcal{E}_{\beta,\gamma,i} \in \mathcal{B}_{\beta,\gamma}} \tau_i D^2\mathcal{E}_{\beta,\gamma,i}(\mathbf{x}_3, \mathbf{x}_4) = 0. \end{aligned} \tag{5}$$

Expressed in a basis (e.g. the u'_i basis used in Definition 4) where the first s basis vectors are chosen to be outside the band kernel, and the remaining $s^2 - s$ basis vectors are chosen from within the band kernel, the column band-space differentials, $D^2\mathcal{E}_{\beta,\gamma,i}$ are 3-tensors of the form shown in Figure 1.

Likewise $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ take the form $(0|\ \mathbf{x}_k\ )$. The 2-tensors $D^2\mathcal{E}_{\beta,\gamma,i}(\mathbf{x}_k)$ can then be represented by matrices of the form:

$$D^2\mathcal{E}_{\beta,\gamma,i}(\mathbf{x}_k) = \left[\begin{array}{c|c} S_{k,i} & R_{k,i} \\ \hline R_{k,i}^\top & 0 \end{array}\right] \tag{6}$$

where $R_{k,i}$ is a random $s \times s^2 - s$ matrix and $S_{k,i}$ is a random symmetric $s \times s$ matrix. Removing the redundant degrees of freedom we have the system of $2s$ equations in s variables:

$$\begin{aligned} \sum_{i=1}^{s} \tau_i R_{1,i} {\mathbf{x}_2}^\top = 0, \\ \sum_{i=1}^{s} \tau_i R_{3,i} {\mathbf{x}_4}^\top = 0. \end{aligned} \tag{7}$$

This has a nontrivial solution precisely when the following $2s \times s$ matrix has nonzero column corank:

$$M = \left[\begin{array}{cccc} | & | & & | \\ R_{1,1}{\mathbf{x}_2}^\top & R_{1,2}{\mathbf{x}_2}^\top & \dots & R_{1,s}{\mathbf{x}_2}^\top \\ | & | & & | \\ \hline | & | & & | \\ R_{3,1}{\mathbf{x}_4}^\top & R_{3,2}{\mathbf{x}_4}^\top & \dots & R_{3,s}{\mathbf{x}_4}^\top \\ | & | & & | \end{array}\right] \tag{8}$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

This is a random matrix over $k = \mathbb{F}_q$, which has nonzero column corank with probability approximately $\frac{1}{(q-1)q^s}$, for practical parameters.

To verify that the conditions given in the theorem are sufficient to establish the randomness of the matrix M, we can give the following explicit expression for the matrix M, which is most easily derived by applying the product rule for the discrete differential to Definition 3:

$$M = \left[\begin{array}{cccc} Dp_1(\mathbf{x}_1,\mathbf{x}_2) & Dp_{s+1}(\mathbf{x}_1,\mathbf{x}_2) & \cdots & Dp_{s^2-s+1}(\mathbf{x}_1,\mathbf{x}_2) \\ Dp_2(\mathbf{x}_1,\mathbf{x}_2) & Dp_{s+2}(\mathbf{x}_1,\mathbf{x}_2) & \cdots & Dp_{s^2-s+2}(\mathbf{x}_1,\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ Dp_s(\mathbf{x}_1,\mathbf{x}_2) & Dp_{2s}(\mathbf{x}_1,\mathbf{x}_2) & \cdots & Dp_{s^2}(\mathbf{x}_1,\mathbf{x}_2). \\ \hline Dp_1(\mathbf{x}_3,\mathbf{x}_4) & Dp_{s+1}(\mathbf{x}_3,\mathbf{x}_4) & \cdots & Dp_{s^2-s+1}(\mathbf{x}_3,\mathbf{x}_4) \\ Dp_2(\mathbf{x}_3,\mathbf{x}_4) & Dp_{s+2}(\mathbf{x}_3,\mathbf{x}_4) & \cdots & Dp_{s^2-s+1}(\mathbf{x}_3,\mathbf{x}_4) \\ \vdots & \vdots & \ddots & \vdots \\ Dp_s(\mathbf{x}_3,\mathbf{x}_4) & Dp_{2s}(\mathbf{x}_3,\mathbf{x}_4) & \cdots & Dp_{s^2}(\mathbf{x}_3,\mathbf{x}_4) \end{array}\right] \tag{9}$$

Combining the results of Theorems 2 and 3, we find that for each choice of the vectors $\mathbf{x}_k$, there is a column band-space map among the solutions of Equation (3) with probability approximately $\frac{1}{(q-1)^2q^{2s}}$ for even characteristic and $\frac{1}{(q-1)^2q^s}$ for odd characteristic. Equation (3) is a system of $2s^2$ equations in $2s^2$ variables; one might expect it to generally have a 0-dimensional space of solutions. In some cases, however, there are linear dependencies among the equations, due to the fact that the $D^2\mathcal{E}_i$ are symmetric tensors. In even characteristic, we get 4 linear dependencies: $D^2\mathcal{E}_i(\mathbf{x}_1,\mathbf{x}_2)(\mathbf{x}_1) = 0$, $D^2\mathcal{E}_i(\mathbf{x}_1,\mathbf{x}_2)(\mathbf{x}_2) = 0$, $D^2\mathcal{E}_i(\mathbf{x}_3,\mathbf{x}_4)(\mathbf{x}_3) = 0$, and $D^2\mathcal{E}_i(\mathbf{x}_3,\mathbf{x}_4)(\mathbf{x}_4) = 0$, and an additional linear dependency when we reduce the number of independent vectors to 3 by setting $\mathbf{x}_1 = \mathbf{x}_4$: $D^2\mathcal{E}_i(\mathbf{x}_1,\mathbf{x}_2)(\mathbf{x}_3) + D^2\mathcal{E}_i(\mathbf{x}_3,\mathbf{x}_4)(\mathbf{x}_2) = 0$, resulting in a 5-dimensional space of solutions. In characteristic 3, reducing the number of independent vectors to 2 results in 2 linear dependencies among the equations: e.g. setting $\mathbf{x}_1 = \mathbf{x}_2$ and $\mathbf{x}_3 = \mathbf{x}_4$, we have $D^2\mathcal{E}_i(\mathbf{x}_1,\mathbf{x}_2)(\mathbf{x}_1) = 0$ and $D^2\mathcal{E}_i(\mathbf{x}_3,\mathbf{x}_4)(\mathbf{x}_3) = 0$. In higher characteristic, there are no linear dependencies imposed on the equations by setting $\mathbf{x}_1 = \mathbf{x}_2$ and $\mathbf{x}_3 = \mathbf{x}_4$.

For characteristic 2, finding the expected 1-dimensional space of band-space solutions in a 5-dimensional space costs $q^4 + q^3 + q^2 + q + 1$ rank operations, which in turn cost $(s^2)^\omega$ field operations, where $\omega \approx 2.373$ is the linear algebra constant. Likewise, for characteristic 3, finding the expected 1-dimensional space of band-space solutions in a 2-dimensional space costs $q+1$ rank operations. Thus the total cost of finding a column band-space map using our variant of MinRank is approximately $q^{2s+6}s^{2\omega}$ for charactersitic 2, $q^{s+3}s^{2\omega}$ for characteristic 3, and $q^{s+2}s^{2\omega}$ for higher characteristic.

6 Complexity of Invariant Attack

The detection of a low rank induced bilinear form $D^2\mathcal{E}(x)$ already constitutes a distinguisher from a random system of equations. Extending this calculation to

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

a full key recovery requires further use of the differential invariant structure of the public key.

First, note that U is not a critical element of the scheme. If A is a random matrix of quadratic forms and B and C are random matrices of linear forms, so are $A \circ U$, $B \circ U$ and $C \circ U$ for any full rank map U. Thus, since clearly $T \circ \phi(AB||AC) \circ U = T \circ \phi((A \circ U)(B \circ U)||(A \circ U)(C \circ U))$, we may absorb the action of U into A, B, and C, and consider the public key to be of the form:

$$P(\mathbf{x}) = T \circ \phi(AB||AC)(\mathbf{x}).$$

Next, consider a trilinear form $D^2\mathcal{E}$ in the band space generated by $\mathcal{B}_{\beta,\gamma}$. Since the coefficients of $D^2\mathcal{E}$ are products of coefficients of A and coefficients of an element of $Im(B||C)$, both of which are uniform i.i.d., there is a change of basis M in which $D^2\mathcal{E}$ has the form in Figure 1 and the nonzero coefficients are uniform i.i.d.

Consider $D^2\mathcal{E}(\mathbf{x}_1)$ and $D^2\mathcal{E}(\mathbf{x}_2)$ for $\mathbf{x}_1, \mathbf{x}_2$ in the band kernel corresponding to $\mathcal{B}_{\beta,\gamma}$. Being maps from the same band space, there is a basis in which both $D^2\mathcal{E}(\mathbf{x}_1)$ and $D^2\mathcal{E}(\mathbf{x}_2)$ have the form in Figure 2. Thus, with high probability for $s \geq 2$, the kernels of both maps are contained in the corresponding band kernel, $\mathcal{B}_{\beta,\gamma}$, and $\text{span}(\ker(D^2\mathcal{E}(\mathbf{x}_1) \cup \ker(D^2\mathcal{E}(\mathbf{x}_2)) = \mathcal{B}_{\beta,\gamma}$.

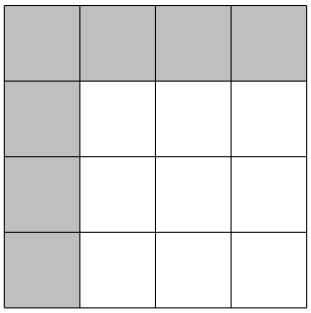


Fig. 2. Structure of the bilinear forms induced by cubic maps in the same band space.

Remark 1 *Here we have utilized a property which explicitly distinguishes differential invariant structure from rank structure.*

Given the basis for an $s^2 - s$ dimensional band kernel $\mathcal{BK}$, we may choose a basis $\{v_1, \ldots, v_s\}$ for the subspace of the dual space vanishing on $\mathcal{BK}$. We can

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

also find a basis $\mathcal{E}_{v_1}, \ldots, \mathcal{E}_{v_s}$ for the band space itself by solving the linear system

$$\begin{aligned}
\sum_{\mathcal{E}_i} \tau_i D^2 \mathcal{E}_i(\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}) &= 0, \\
\sum_{\mathcal{E}_i} \tau_i D^2 \mathcal{E}_i(\mathbf{x}_{21}, \mathbf{x}_{22}, \mathbf{x}_{23}) &= 0, \\
&\vdots = \vdots \\
\sum_{\mathcal{E}_i} \tau_i D^2 \mathcal{E}_i(\mathbf{x}_{t1}, \mathbf{x}_{t2}, \mathbf{x}_{t3}) &= 0,
\end{aligned}$$

where $t \approx 2s^2$ and $\mathbf{x}_{ij}$ is in the band kernel.

Since the basis $\mathcal{E}_{v_1}, \ldots, \mathcal{E}_{v_s}$ is in a single band space, there exists an element $\begin{bmatrix} b'_1 & \cdots & b'_s \end{bmatrix}^\top \in ColSpace(B||C)$ and two matrices Ω_1 and Ω_2 such that

$$\Omega_1 A \left(\Omega_2 \begin{bmatrix} b'_1 \\ \vdots \\ b'_s \end{bmatrix} \right) =: A' \left(\begin{bmatrix} v_1 \\ \vdots \\ v_s \end{bmatrix} \right) = \begin{bmatrix} \mathcal{E}_{v_1} \\ \vdots \\ \mathcal{E}_{v_s} \end{bmatrix}.$$

Solving the above system of equations over $\mathbb{F}_q[x_1, \ldots, x_{s^2}]$ uniquely determines A' in $\mathbb{F}_q[x_1, \ldots, x_{s^2}]/\langle v_1, \ldots, v_s \rangle$. To recover all of A', note that the above system is part of an equivalent key

$$\mathcal{F} = T' \circ A'(B'||C')$$

where $\begin{bmatrix} v_1 & \cdots & v_s \end{bmatrix}^\top$ is the first column of B'.

Applying T'^{-1} to both sides and inserting the information we know we may construct the system

$$A'(B'||C') = T'^{-1}\mathcal{F} \tag{10}$$

Solving this system of equations modulo $\langle v_1, \ldots, v_s \rangle$ for B', C' and T'^{-1} we can recover a space of solutions, which we will restrict by arbitrarily fixing the value of T'^{-1}. Note that the elements of T'^{-1} are constant polynomials, and therefore $T'^{-1} (\mathrm{mod}\, \langle v_1, \ldots, v_s \rangle)$ is the same as T'^{-1}. Thus, for any choice of T'^{-1} in this space, the second column of $T'^{-1}\mathcal{F}$ is a basis for a band space. Moreover, the elements $v'_{s+1}, \ldots, v'_{2s}$ of the second column of $B' (\mathrm{mod}\, \langle v_1, \ldots, v_s \rangle)$ are the image, modulo $\langle v_1, \ldots, v_s \rangle$, of linear forms vanishing on the corresponding band kernel. Therefore, the intersection $\bigcap_{i=1}^{s} \ker(v_i) \cap \bigcap_{i=s+1}^{2s} \ker(v'_i)$ is the intersection $\mathcal{BK}_2 \cap \mathcal{BK}_1$ of the band kernels of our two band spaces.

We can reconstruct the full band kernel of this second band space using the same method we used to obtain our first band kernel: We take a map $\mathcal{E}_2$ from the second column of $T'^{-1}\mathcal{F}$, and two vectors x_a and x_b from $\mathcal{BK}_2 \cap \mathcal{BK}_1$, and we compute $\mathcal{BK}_2 = \mathrm{span}(\ker(D^2\mathcal{E}_2(\mathbf{x}_a) \cup \ker(D^2\mathcal{E}_2(\mathbf{x}_b))$. We can now solve for the second column of B', $\begin{bmatrix} v_{s+1} & \cdots & v_{2s} \end{bmatrix}^\top$, uniquely over $\mathbb{F}_q[x_1, \ldots, x_{s^2}]$ (NOT modulo $\langle v_1, \ldots, v_s \rangle$) by solving the following system of linear equations:

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

$$\begin{aligned} v_i \equiv v_i' (\mathrm{mod}\, \langle v_1, \ldots, v_s \rangle) \\ v_i(\mathbf{x}_1) = 0 \\ v_i(\mathbf{x}_2) = 0 \\ \vdots = \vdots \\ v_i(\mathbf{x}_{s^2-s}) = 0 \end{aligned}$$

where $i = s+1, \ldots, 2s$, and $(\mathbf{x}_1, \ldots, \mathbf{x}_{s^2-s})$ is a basis for $\mathcal{BK}_2$. We can now solve for A' (again, uniquely over $\mathbb{F}_q[x_1, \ldots, x_{s^2}]$) by solving:

$$A' \left(\begin{bmatrix} v_1 \\ \vdots \\ v_s \end{bmatrix} \right) \equiv \begin{bmatrix} \mathcal{E}_{v_1} \\ \vdots \\ \mathcal{E}_{v_s} \end{bmatrix} (\mathrm{mod}\, \langle v_1, \ldots, v_s \rangle)$$

$$A' \left(\begin{bmatrix} v_{s+1} \\ \vdots \\ v_{2s} \end{bmatrix} \right) \equiv \begin{bmatrix} \mathcal{E}_{v_{s+1}} \\ \vdots \\ \mathcal{E}_{v_{2s}} \end{bmatrix} (\mathrm{mod}\, \langle v_{s+1}, \ldots, v_{2s} \rangle)$$

where $\begin{bmatrix} \mathcal{E}_{v_{s+1}} & \cdots & \mathcal{E}_{v_{2s}} \end{bmatrix}^\top$ is the second column of $T'^{-1}\mathcal{F}$. This allows us to solve equation 10 for the rest of B' and C', completing the attack.

The primary cost of the attack involves finding the band space map. The rest of the key recovery is additive in complexity and dominated by the band space map recovery; thus, the total complexity of the attack is of the same order as band space map recovery. Hence, the cost of private key extraction is approximately $q^{2s+6}s^{2\omega}$ for characteristic 2, $q^{s+3}s^{2\omega}$ for characteristic 3, and $q^{s+2}s^{2\omega}$ for higher characteristic. We note that with these parameters we can break full sized instances of this scheme with parameters chosen for the 80-bit and 100-bit security levels via the criteria presented in [14].

Specifically, our attack breaks CubicABC($q = 127$,$s = 7$), designed for 80-bit security, in approximately 2^{76} operations (or around 2^{80} if one pessimistically uses $\omega = 3$ as the linear algebra constant). More convincingly, our attack completely breaks CubicABC($q = 127$,$s = 8$), designed for 100-bit security, in approximately 2^{84} operations (or 2^{88} for $\omega = 3$). Furthermore, the attack is fully parallelizable and requires very little memory; hence, the differential invariant attack is far more efficient than algebraic attacks, the basis for the original security estimation. Thus, the security claims in [14] are clearly unfounded; in fact, the cubic version of the scheme, whose security was claimed to be closely related to an NP-complete problem, is actually less secure than the quadratic case.

We can explain this dramatic discrepancy on the fact that the parameters in [14] are derived by assuming that the algebraic attack is the most effective. In the case of the quadratic ABC scheme, for the proposed parameters, the attack of [13] was slower than the algebraic attack, though asymptotically faster. In the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

case of the Cubic scheme, the attack is actually more efficient, in asymptotics as well as for practical parameters.

7 Experiments

Using SAGE [15], we performed some minrank computations on small scale variants of the Cubic ABC scheme. The computations were done on a computer with a 64 bit quad-core Intel i7 processor, with clock cycle 2.8 GHz. We were interested in verifying our complexity estimates on the most costly step in the attack, the MinRank instance, rather than the full attack on the ABC scheme. Given as input the finite field size q, and the scheme parameter s, we computed the average number of vectors v required to be sampled in order for the rank of the 2-tensor $D^2\mathcal{E}(v)$ to fall to $2s$. As explained in Section 5, when the rank falls to this level, we have identified the subspace differential invariant structure of the scheme and can exploit this structure to attack the scheme. Our results for odd q are given in Table 1.

	$s=3$	$(q-1)^2q^s$	$s=4$	$(q-1)^2q^s$	$s=5$	$(q-1)^2q^s$
$q=3$	14.75	108	333	324	952	972
$q=5$	378	2000	9986	10000		
$q=7$	1688	12348	72951	86436		
$q=9$	606	46656				
$q=11$	13574	133100				

Table 1. Average number of vectors needed for the rank to fall to $2s$ (for odd q)

For higher values of q and s the computations took too long to produce sufficiently many data points and obtain meaningful results with SAGE. When q is odd, our analysis predicted the number of vectors needed would be on the order of $(q-1)^2q^s$. Table 1 shows the comparison between our experiments and the expected value. We see that for $s=3$, the rank fell quicker than expected, while for $s>3$ the results are quite near the predicted value. This is because when $s=3$ our complexity estimates given in Section 5 are simply not accurate enough, which happens for small values of q and/or s.

For even q, we also ran some experiments. We found that for $s=3$ and $q=2,4$, or 8, with high probability only a single vector was needed before the rank fell to $2s$. For $s=4$ and $s=5$, the computations were only feasible in SAGE for $q=2$. The average number of vectors needed in the $s=4$ case was 244, with the expected value being $(q-1)^2q^{2s}=256$. With $s=5$, the average number in our experiments was 994 (although the number of trials was small), with the expected value 1024. For higher values of q and s the computations took too long to obtain meaningful results.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Key Recovery Attack on Cubic Simple Matrix Encryption."
Paper presented at Selected Areas in Cryptography (SAC 2016), St. Johns, Newfoundland, Canada. August 10, 2016 - August 12, 2016.

8 Conclusion

The ABC schemes are very interesting new ideas for multivariate public key schemes. Essentially all of MPKC can be bisected into big field schemes, utilizing the structure of an extension of the field used for public calculations, and small field schemes which require no such extension. (For the purpose of this comment we consider "medium" field schemes to be big field schemes.)

The ABC cryptosystems present a fundamentally new structure for the development of schemes. In fact, if we consider the structure of simple algebras over the public field (which are surely the only such structures we should consider for secure constructions) then "big field" and "big matrix algebra" complete the picture of possible large structure schemes.

It is interesting to note that the authors provide in [14] a heuristic security argument for the scheme and, as reinforced in the first presentation of the scheme at [16], suggest that with some work the scheme may be able to be shown provably secure. The idea behind their argument is at least somewhat reasonable, if not precise. Their argument essentially amounts to the following: every cubic polynomial in the public key is in the ideal generated by the quadratic forms in A under a certain basis; thus, one might expect the public key to contain a subset of the information one would obtain by applying one step of a Gröbner basis algorithm such as F4, see [17].

Unfortunately, this analysis is not very tight. In fact, we exploit the subspace differential invariant structure inherent to the ABC methodology to show that for odd characteristic the cubic scheme is less secure than its quadratic counterpart. We may therefore conclude that any attempt at a secure cubic "big matrix algebra" scheme must rely on the application of modifiers. The challenge, then, is to construct such a scheme which is still essentially injective for the purpose of encryption. Schemes such as this one can never compete with the secure multivariate options for digital signatures we already know.

We are thus left with the same lingering question that has been asked for the last two decades: Is secure multivariate encryption possible? Currently there is a small list of candidates none of which has both been extensively reviewed and has existed for longer than a few years. If we are to discover a secure multivariate encryption scheme with a convincing security proof or some other security metric, it will require some new techniques and new science. Only time will tell.

References

1. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Sci. Stat. Comp. **26, 1484** (1997)
2. Chen, L., Jordan, S., Liu, Y., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography. NISTIR 8105 (2016) http://dx.doi.org/10.6028/NIST.IR.8105.
3. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. EUROCRYPT 1999. LNCS **1592** (1999) 206–222

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

4. Patarin, J., Goubin, L., Courtois, N.: C $^{*}_{-+}$ and HM: Variations around two schemes of T.Matsumoto and H.Imai. Asiacrypt 1998, Springer **1514** (1998) 35–49
5. Patarin, J., Courtois, N., Goubin, L.: Quartz, 128-bit long digital signatures. In Naccache, D., ed.: CT-RSA. Volume 2020 of Lecture Notes in Computer Science., Springer (2001) 282–297
6. Petzoldt, A., Bulygin, S., Buchmann, J.: Cyclicrainbow - a multivariate signature scheme with a partially cyclic public key. In Gong, G., Gupta, K.C., eds.: INDOCRYPT. Volume 6498 of Lecture Notes in Computer Science., Springer (2010) 33–48
7. Petzoldt, A., Chen, M., Yang, B., Tao, C., Ding, J.: Design principles for hfev-based multivariate signature schemes. In Iwata, T., Cheon, J.H., eds.: Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I. Volume 9452 of Lecture Notes in Computer Science., Springer (2015) 311–334
8. Ding, J., Yang, B.Y.: Degree of regularity for hfev and hfev-. [18] 52–66
9. Goubin, L., Courtois, N.: Cryptanalysis of the ttm cryptosystem. In Okamoto, T., ed.: ASIACRYPT. Volume 1976 of Lecture Notes in Computer Science., Springer (2000) 44–57
10. Tsujii, S., Gotaishi, M., Tadaki, K., Fujita, R.: Proposal of a signature scheme based on sts trapdoor. In Sendrier, N., ed.: PQCrypto. Volume 6061 of Lecture Notes in Computer Science., Springer (2010) 201–217
11. Porras, J., Baena, J., Ding, J.: Zhfe, a new multivariate public key encryption scheme. [16] 229–245
12. Tao, C., Diene, A., Tang, S., Ding, J.: Simple matrix scheme for encryption. [18] 231–242
13. Moody, D., Perlner, R.A., Smith-Tone, D.: An asymptotically optimal structural attack on the ABC multivariate encryption scheme. [16] 180–196
14. Ding, J., Petzoldt, A., Wang, L.: The cubic simple matrix encryption scheme. [16] 76–87
15. Developers, T.S.: SageMath, the Sage Mathematics Software System (Version 7.3). (2016) `http://www.sagemath.org`.
16. Mosca, M., ed.: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Volume 8772 of Lecture Notes in Computer Science., Springer (2014)
17. Faugere, J.C.: A new efficient algorithm for computing grobner bases (f4). Journal of Pure and Applied Algebra **139** (1999) 61–88
18. Gaborit, P., ed.: Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings. In Gaborit, P., ed.: PQCrypto. Volume 7932 of Lecture Notes in Computer Science., Springer (2013)

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Towards Actionable Mission Impact Assessment in the Context of Cloud computing

Xiaoyan Sun
Pennsylvania State University
University Park, PA 16802,
USA
xzs5052@ist.psu.edu

Anoop Singhal
National Institute of Standards
and Technology
Gaithersburg, MD 20899, USA
anoop.singhal@nist.gov

Peng Liu
Pennsylvania State University
University Park, PA 16802,
USA
pliu@ist.psu.edu

ABSTRACT

Today's cyber-attacks towards enterprise networks often undermine and even fail the mission assurance of victim networks. Mission cyber resilience (or active cyber defense) is critical to prevent or minimize negative consequences towards missions. Without effective mission impact assessment, mission cyber resilience cannot be really achieved. However, there is an overlooked gap between mission impact assessment and cyber resilience due to the non-mission-centric nature of current research. This gap is even widened in the context of cloud computing. The gap essentially accounts for the weakest link between missions and attack-resilient systems, and also explains why the existing impact analysis is not really actionable. This paper initiates efforts to bridge this gap, by developing a novel graphical model that interconnects the mission dependency graphs and cloud-level attack graphs. Our case study shows that the new cloud-applicable model is able to bridge the gap between mission impact assessment and cyber resilience. As a result, it can significantly boost the cyber resilience of mission critical systems.

1. INTRODUCTION

Due to the increasing severity of cyber-attacks, mission assurance entails critical demands of active cyber defense and cyber resilience more than ever. Mission cyber resilience or active cyber defense means capabilities to make prioritized, proactive and resource-constraint-aware recommendations on taking cyber defense actions, including network and host hardening actions, quarantine actions, adaptive MTD (Moving Target Defense) actions, roll-back actions, repair and regeneration actions. Due to the fundamental necessity and importance of situational awareness to decision making, cyber situational awareness plays a critical role in achieving mission cyber resilience. Especially, mission cyber resilience cannot be really achieved without impact assessment. That is, knowing which mission and how a mission is impacted by an attack is the key for making correct resilience decisions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
SafeConfig'16, October 24, Vienna, Austria.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

However, there is actually a largely overlooked gap between mission impact assessment and cyber resilience, though both mission impact assessment and attack-resilient systems have been extensively researched in the literature: 1) Despite extensive research on attack-resilient survivable systems and networks [1], most if not all existing cyber resilience techniques are unfortunately not mission-centric. Lack of mission models and mission dependency analysis is a common limitation of existing attack resilience techniques. Without mission dependency analysis, existing cyber resilience techniques cannot quantify the effectiveness of the recommended cyber response actions in terms of mission goals, and hence cannot convincingly justify the superiority of the recommended response actions. 2) From another aspect, despite extensive research on mission impact assessment, mission impact assessment results cannot be automatically used to make mission-centric recommendations on taking cyber response actions. This gap is even widened in the context of cloud computing. In public cloud, each enterprise network has its own missions. These missions are usually expected to be independent and isolated from each other. However, multi-step attacks may penetrate the boundaries of individual enterprise networks from the same cloud, and thus impact missions of multiple enterprise networks. That is, attacks that happen in one enterprise network may be able to affect missions of another enterprise network in the same cloud. Therefore, mission impact should be re-assessed in cloud environment.

Hence, lack of automation tools in associating missions with attack-resilient systems is a weakest link in achieving cyber resilience. Without such association, existing mission impact analysis results are not really actionable: it's difficult to find out why and how a mission has been impacted. Since bridging this gap may significantly boost the cyber resilience of mission critical systems, how to bridge this gap is a very important problem.

Therefore, the primary objective of this paper is to take the first steps towards systematically bridging the critical gap between mission impact assessment and cyber resilience in the context of cloud computing. We aim to model the mission impact process and enable the automatic reasoning of this process. To achieve this goal, we identified and addressed the following challenges:

First, it is very challenging to envision a never-seen-before graphical model that can integrate mission dependency graphs and cloud-level attack graphs in such a way that can effectively bridge the gap between existing mission impact analysis results and attack-graph-based active cyber defense. No

Liu, Peng; Singhal, Anoop; Sun, Xiaoyan.
"Towards Actionable Mission Impact Assessment in the Context of Cloud Computing."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

graphical model has yet been proposed to bridge this gap, though two schools of thoughts have been respectively developed on mission impact analysis and attack-graph-based active cyber defense.

Second, a cloud environment gives rise to new challenges in bridging the gap. Cloud services such as Infrastructure as a Service (IaaS), make attack graphs more complicated and harder to get analyzed. Conventional attack graphs cannot capture the stealthy information flows introduced by certain cloud features. Attackers could leverage the hidden security vulnerabilities caused by inappropriate cloud management to launch zero-day attacks.

The significance of this paper's contributions is two-fold: 1) We have developed a novel graphical model, the mission impact graph model, to systematically bridges the critical gap between impact assessment and cyber resilience. Bridging this gap significantly boosts the cyber resilience of mission critical systems; 2) To the best of our knowledge, this is the first work that investigates the mission impact assessment problem by considering the special features in cloud computing environment; 3) We have extended the attack graph generation tool MulVAL [9] to enable logical reasoning of mission impact assessment and automatic generation of mission impact graphs.

2. RELATED WORK

A literature review is firstly performed to disclose the mismatch between mission impact assessment and cyber resilience: 1) the formal models used by the existing mission impact assessment techniques cannot be directly used by the existing attack-resilient system and network designs; 2) lack of mission models and mission dependency analysis is a common limitation of existing cyber resilience techniques.

Mission impact assessment. In the past decade or so, extensive research has been conducted on modeling the mission dependencies to help facilitate computer-assisted analysis of current missions. The existing mission-oriented impact assessment techniques can be classified into four categories: 1) mission impact assessment through use of ontology based data collection. The basic idea is to create the ontology of mission dependencies. For example, the Cyber Assets to Mission and Users (CAMUs) approach [2] assumes that a cyber asset provides a cyber capability that in turn supports a mission. Their approaches mine existing logs and configurations, such as those from LDAP, NetFlow, FTP, and UNIX to create these mission-asset mappings; 2) mission impact assessment through use of dependency graphs [4, 5]. The basic idea is the use of mission dependency graphs for cyber impact assessment and a hierarchical (time-based) approach to mission modeling and assessment; 3) mission impact assessment through use of mission thread modeling [6]. The basic idea is to leverage mission metrics supported by resource model and value model; 4) mission impact assessment through use of Yager's aggregators [3]. The basic idea is to utilize a tree-based approach to calculate the impact of missions. The mission tree is a tree-structure that utilizes Yager's aggregators [7] to intelligently aggregate the damage of assets to calculate the impact on each individual mission.

Cyber resilience and active cyber defense. Since 2000, a tremendous amount of research has been conducted on how to make systems and networks resilient to cyber-attacks. For example, the two volumes of DARPA Information Survivability Conference and Exposition proceedings described

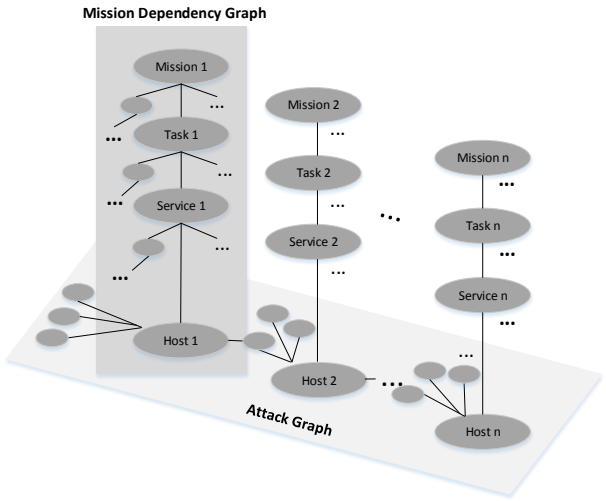


Figure 1: The Mission Dependency Graph and Attack Graph.

the design, implementation and evaluation of the first set of survivable and attack-resilient systems and networks [1]. The cyber response actions adopted in these systems include replication actions, honeypot actions, software diversification actions, dynamic quarantine actions, adaptive defense actions, roll-back actions, proactive and reactive recovery actions. Since then, a variety of cyber response actions have emerged, including migration actions, regeneration actions, MTD actions, decoy actions, CFI (control flow integrity) actions, ASLR (Address Space Layout Randomization) actions, IP randomization actions, N-variant defense actions, and software-defined network virtualization actions.

3. OUR APPROACH

In this paper, we aim to bridge the gap between mission impact assessment and cyber resilience.

On the side of mission impact assessment, different types of mission dependency graphs have been developed to associate missions with component tasks and assets. As shown in Figure 1, the status of assets (hosts, virtual machines, etc.) will generate direct impact towards missions through dependency relations. In current literature, such dependency relations among assets, tasks, and missions are usually very loose and not well defined. As a result, the corresponding mission impact assessment is also inaccurate. In addition, without considering the possibility of multi-step attacks caused by combinations of vulnerabilities, the mission impact assessment is usually not sufficiently comprehensive. For example, in Figure 1, assuming *mission 1* and *mission n* depends on *host 1* and *host n* respectively, if *host 1* is compromised, then *mission 1* will be impacted and *mission n* will not. However, if *host 1* can be used as a stepping-stone to compromise *host n*, then *mission n* has the possibility of being impacted as well. Therefore, with only mission dependency graph, it is not sufficient to perform accurate and comprehensive mission impact assessment.

On the side of cyber resilience, attack graphs have become mature techniques for analyzing the causality relationships between vulnerabilities and exploitations. As in Figure 1, by analyzing the vulnerabilities existing in the network, attack graphs are able to generate potential attack paths that show a sequence of attack steps (from *host 1* to *host n*).

Liu, Peng; Singhal, Anoop; Sun, Xiaoyan.
"Towards Actionable Mission Impact Assessment in the Context of Cloud Computing."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This capability enables security admins to proactively analyze the influence of some security operations towards the potential attack paths. For example, security admins could check how potential attack paths would be changed if they patch a vulnerability. However, the traditional attack graph has two limitations. First, it is not mission-centric. The attack graph is able to generate potential attack paths through logical reasoning, but it lacks the capability to reason potential impacts towards missions. Second, traditional attack graphs do not consider potential attacks enabled by the cloud environment.

Therefore, considering the respective capabilities and disadvantages of mission dependency graphs and attack graphs, this paper proposes to develop a logical graphical model, called mission impact graph, to integrate mission dependency graphs and cloud-level attack graphs. Our approach contains three steps. First, there exist essential semantic gaps between mission dependency graphs and attack graphs. We identify the semantic gaps and unify the representation of nodes and edges. This makes interconnecting mission dependency graphs and attack graphs feasible. Second, to bridge the gap inside a cloud environment, we extend traditional attack graphs into cloud-level attack graphs. The cloud-level attack graphs are incorporated into new mission impact graph. Third, we implement a set of interaction rules in MulVAL [8, 9] to enable automatic generation of logical mission impact graph.

4. THE SEMANTIC GAP BETWEEN THE ATTACK GRAPH AND THE MISSION DEPENDENCY GRAPH

Generally speaking, a mission dependency graph is a mathematical abstraction of assets, services, mission steps (also known as tasks) and missions, and all of their dependencies [6]. A mission dependency graph has five types of nodes, including assets, services, tasks, missions and logical dependency nodes. The logical dependency nodes are basically AND-nodes and OR-nodes that represent logical dependencies among other nodes. The AND-node represents that a parent nodes depends on all of its children nodes. The OR-node denotes that a parent node depends on at least one of its children nodes. For example, a successful task may depend on all of the supporting services being functional, while a complete mission could require only one of its tasks being fulfilled. Edges in a mission dependency graph represent the interdependencies existing among nodes.

As for the attack graph, it usually shows the potential attack steps leading to an attack goal. Several different types of attack graphs have been developed, such as state enumeration attack graphs [10–12] and dependency attack graphs [13–15]. This paper uses the dependency attack graph for analysis. Figure 2 is part of a simplified attack graph. A traditional attack graph generated by MulVAL is composed of two types of nodes, fact nodes (including primitive fact nodes and derived fact nodes) and derivation nodes (also known as rule nodes). Primitive fact nodes (denoted with rectangles in Figure 2) present objective conditions of the network, such as the network, host, and vulnerability information. Derived fact nodes (denoted with diamonds) are the facts inferred by applying the derivation rule. Each derivation node (denoted with ellipse) represents the application of a derivation rule. The derivation rules are implemented as interaction rules in MulVAL. Simply put, one or more fact nodes could be the preconditions of a derivation node, while the derived fact node is the post-condition of the derivation node. For example, in Figure 2, if node 4 "the attacker has access to the server", node 5 "the server provides a service with an application" and node 6 "the application has a vulnerability" are all satisfied, then the rule in node 7 will take effect and make node 8 become true. That is, attacker is able to execute arbitrary code on the server.

Mission dependency graphs and traditional attacks graphs have the following semantic gaps:

1) The meaning of nodes differs. In a mission dependency graph, a node denotes an entity, such as an asset, a service, a task, or a mission. The node does not specify the status of the entity. In a traditional attack graph, a node represents a statement, be it a rule or a fact. For example, a primitive fact node could be "the web server provides OpenSSL service" or "the openssl program has a vulnerability called CVE-2008-0166". A rule node could be "the remote exploit of a server program could happen".

2) The meaning of edges differs. In a mission dependency graph, the edges represent general interdependencies among nodes, and do not specify concrete dependency types. The logical relations are specially denoted with AND and OR nodes. In a traditional attack graph, directed edges represent the causality relationship among nodes. One or more fact nodes could cause a derivation node to take effect, which further enables a derived fact node.

3) The representation of logical relations among nodes differs. In a mission dependency graph, the logical relations are represented specifically with AND and OR nodes. In traditional attack graph, the logical relations are not provided explicitly, but are implied in the graph structure: derivation nodes (rule nodes) imply AND relations and derived fact nodes imply OR relations. That is, fact nodes that serve as preconditions of a derivation node have AND relations, while derivation nodes leading to a derived fact node have OR relations. The underlying principle is that all of the preconditions have to be satisfied to enable a derivation rule, while a derived fact node can become true as long as one rule is satisfied.

5. INCORPORATING CLOUD-LEVEL ATTACK GRAPHS

In the public cloud, each enterprise network can generate its own individual attack graph by scanning hosts and virtual machines in the network. These individual graphs may not be complete because new attack paths enabled by the cloud environment could be missed. Therefore, a cloud-level attack graph is needed to capture potential missing attacks by taking some features of public cloud into consideration, such as virtual machine image sharing and virtual machine co-residency. Hence, [16] proposed the construction of cloud-level attack graphs. A cloud-level attack graph contains three levels: virtual machine level, virtual machine image level, and host level. The virtual machine level mainly captures the causality relationship between vulnerabilities and potential exploits inside the virtual machines. The virtual machine image level focuses on attacks related to virtual machine images. For example, a virtual machine image may be instantiated by different enterprise networks. As a result, its security holes are also inherited by all the instance vir-

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

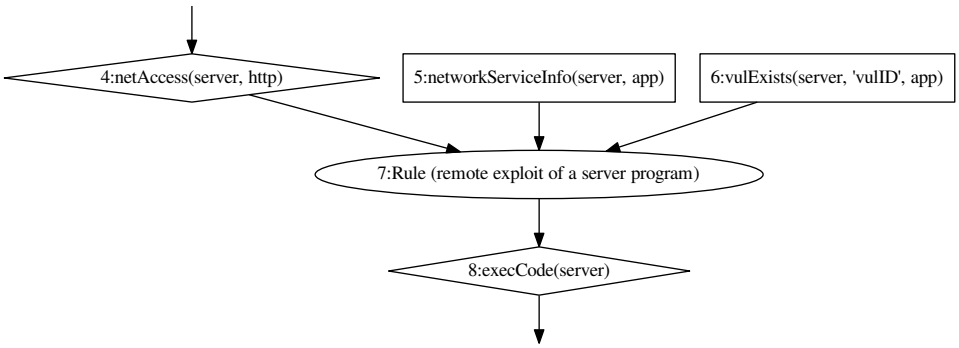


Figure 2: Part of a Simplified Attack Graph.

tual machines. The virtual machine image level is able to reflect such inheritance relationship. The host level mainly captures potential attacks to hosts, including exploits leveraging the virtual machine co-residency relationship.

Therefore, the mission impact graph needs to be extended to incorporate cloud-level attack graphs. The semantics of mission impact graphs remain the same because cloud-level attack graphs have the same semantics as traditional attack graphs. However, the mission impact graph is now composed of two parts: cloud-level attack graph part, and the cloud-applicable mission dependency part. New nodes should be added as derivation nodes and fact nodes to incorporate special features of cloud. To achieve this goal, we crafted a set of Datalog clauses in MulVAL as the primitive facts, derived facts and interaction rules. For the cloud-level attack graph part, new facts and rules are crafted to model virtual machine image vulnerability existence, vulnerability inheritance, backdoor problem, and virtual machine co-residency problem, and so on. For mission dependency part, new rules are added to model the residency dependencies among virtual machines and hosts, service dependencies among virtual machines and services, etc. For example, the residency dependency relationship between a host and the dependent virtual machines can be modeled with the following interaction rule:

```
interaction rule(
   (hostImpact(VM):-
      residencyDepend(Vm, Host),
      HostImpact(Host)),
   rule_desc('An compromised host will impact the dependent
virtual machines')).
```

6. MISSION IMPACT GRAPH

The new graphical model, which is referred to as mission impact graph, is formally defined as follows: 1) It is a directed graph that is composed of two parts: attack graph part, and mission impact part. 2) It contains two kinds of nodes: derivation nodes and fact nodes. Each fact node represents a logical statement. Each derivation node represents an interaction rule that is applied for derivation. There are two types of fact nodes, primitive fact nodes and derived fact nodes. A primitive fact node represents a piece of given information, such as host configuration, vulnerability information, network connectivity, service information, progress status of a mission (e.g. which mission steps are already completed and which are not), and so on. Derived fact nodes are computing results of applying interaction rules iteratively on input facts. 3) The edges in the mission impact graph represent the causality relations among nodes. A derived fact node depends on one or more derivation nodes (which have OR relations); a derivation node depends on one or more fact nodes (which have AND relations).

In mission impact graphs, we need to combine attack graphs and mission dependency graphs by unifying their representation of nodes and edges. It is composed of four steps:

Step 1, the entity nodes in mission dependency graphs are changed into fact nodes in the mission impact graph. The fact nodes describe the status of entities. For example, a service node in the mission dependency graph becomes a fact node showing "the service is disabled" in the mission impact graph, and an asset node may become "attackers can execute arbitrary code on the host", etc. One entity in the mission dependency graph may become a number of fact nodes depending on its possible states.

Step 2, the derivation nodes in the mission impact graph are added to model the causality relationships among fact nodes. The interdependencies among entities such as assets, services, tasks, and missions in the mission dependency graph can be interpreted into specific impact causality rules, which become derivation nodes in mission impact graph. For example, the dependency between a task t and a service s could be interpreted into a rule R: "t will be compromised if s is disabled and t is not completed yet". When node "s is disabled" and node "t is not completed yet" are both satisfied, the derivation node stating rule R will take effect.

Step 3, logical relation nodes in mission dependency graphs are removed, including AND and OR nodes. The logical relations among nodes are implied with graph structure as in the mission impact graph: derivation nodes imply AND, and derived fact nodes imply OR.

Step 4, the fact nodes and derivation nodes are connected with edges to represent direct causality relations, rather than general dependencies.

Finally, to enable automatic generation of mission impact graphs, we extended the MulVAL knowledge base by creating Datalog clauses. Three sets of Datalog clauses are added as primitive facts, derived facts, and interaction rules for the function of mission impact analysis. For primitive facts, we crafted clauses that describe mission-task dependencies, the service types, task service dependencies, and

mission progress status, and so on. The information can be provided by system administrators. For derived facts, we added clauses for the status of missions, tasks, services and assets. To enable the logical reasoning, we created interaction rules to model the causality relationships between pre-conditions and post-conditions. For example, attacks towards servers will impact services that are provided by these servers. The interaction rule describing this causality relationship could be as follows:

```
interaction rule(
   (serviceImpact(Service, H, Perm):-
      hostProvideService(H, Service),
      execCode(H, Perm)),
   rule_desc('An compromised server will impact the dependent
service')).
```

7. CASE STUDY

As shown in Figure 3, our scenario contains three enterprise networks in cloud: A is a start-up company, B is a medical group, and C is a vaccine supplier. In addition to providing existing vaccines, C is also developing a new type of vaccine together with its collaborators. The formula of the new vaccine is still very confidential. For security purposes, C only accepts client requests from trusted IPs. The relationships between A, B and C are: 1) they are on the same cloud; 2) A's webserver and B's database server are two virtual machines that co-reside on the same physical host; In general the cloud provider will host the virtual machines on arbitrary hosts. However, for this attack scenario, we are making this assumption that the two virtual machines are on the same host. This co-residency relationship can be leveraged by attackers. 3) B is a trusted client to C. For simplicity, in Figure 3 we only show servers and workstations scenario involved.

The mission for medical group B, *Bm1*, is to provide medical services to all of its patients. Sample tasks include: *Bt1*, patients make appointments; *Bt2*, access medical records; *Bt3*, order shots or medicine; *Bt4*, administer shots; *Bt5*, update medical records, and so on. The mission for vaccine supplier C, *Cm1*, is to supply vaccines to authorized medical groups, and develop the new type of vaccine with collaborators. The sample tasks include: *Ct1*, ask for login ID and password; *Ct2*, check the ID and password. If the user is medical group, go to *Ct3*. If the user is collaboration partner, go to *Ct4*; *Ct3*, order vaccine; *Ct4*, check and update new vaccine information. *Ct3* and *Ct4* are then composed of a number of subtasks.

In our attack scenario, attacker Mallory (could be a competitor of victim companies) is very interested in the new vaccine, and wants to steal its formula from supplier C. To break into the supplier network, Mallory performs the following attack steps: 1) Mallory compromises A's webserver by exploiting vulnerability CVE-2007-5423 in tikiwiki 1.9.8; 2) Mallory leverages the co-residency relationship to take over B's database server, based on a side channel attack in cloud. Side channel attacks are usually quite difficult, but feasible. Extensive research has been done on side channel attacks [17–19]. 3) B's NFS server has a directory (*/exports*) that is shared by all the servers and workstations inside the company. Normally B's web server should not have write permission to this shared directory. However, due to a configuration error of the NFS export table, the web server is given write permission. Therefore, Mallory uploads a Trojan horse to the shared directory, which is crafted as a management software named *tool.deb*. 4) The innocent Workstation user from B downloads *tool.deb* from NFS server and installs it. This creates an unsolicited connection back to Mallory. 5) The Workstation has access to C's webserver as a trusted client. Mallory then managed to take over it via a brute-force key guessing attack (CVE-2008-0166); 6) Mallory leverages C's webserver as a stepping stone to compromise C's MongoDB database server based on CVE-2013-1892, which allows Mallory successfully steal credential information from an employee login database table; 7) Mallory logins into C's webserver as a collaborator of C, and accesses the project proprietary documentation to collect formula-related vaccine research and development records.

By performing logical reasoning in MulVAL, we generated a mission impact graph for our scenario. Figure 4 shows a part of the graph. MulVAL takes several types of inputs, including vulnerability-scanning report, host configuration, network connectivity, mission-task-service-asset dependencies, and so on. The output is a mission impact graph showing which missions are likely to be affected by considering current status of the networks. The cloud-level mission impact graph is very critical and helpful for understanding potential threats to missions. First, individual mission impact graph may miss important attacks leveraging some features of cloud, and thus generate incorrect evaluation about possible threats to missions. For example, without considering the co-residency relationship between A's webserver and B's database server, B seems to be very safe as the database has no exploitable vulnerability. As a result, mission *Bm1* is viewed as safe. However, our mission impact graph shows that *Bm1* has the possibility of being impacted because the virtual machine co-residency can be leveraged for attack. Second, traditional attack graphs are not mission-centric. Even if attack paths are generated for a network, the potential impact towards mission cannot be assessed without analyzing the dependency relationships among missions, tasks, services and assets. Our mission impact graph is able to show such impact towards missions by considering both attacks and missions.

One function of our mission impact graph is to perform automated "taint" propagation through logical reasoning. Given a "taint", be it a vulnerability, a compromised machine, or a disabled service, the impact of the "taint" can be analyzed through logical reasoning. The mission impact graph is able to reflect affected entities such as assets, services, tasks, and missions. For example, in our case study, if C's webserver is compromised, the mission impact graph will show that task *Ct1* and mission *Cm1* are impacted.

Our mission impact graph is also very important for effective cyber resilience analysis. Propagating the attack-graph-based active cyber defense from the attack graph side to the mission impact side is helpful for performing advanced proactive "what-if" mission impact assessment. Through logical reasoning, impact analysis can be performed all the way from inside a machine to a mission. Given the input information, attack graphs can predict the potential attack paths and identify possibly to-be-affected assets. Mission impact graph extends attack graphs in a way that the prediction of potential attack paths directly enables the prediction of potential mission impact. Therefore, we can perform proactive "what-if" mission impact analysis by changing the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

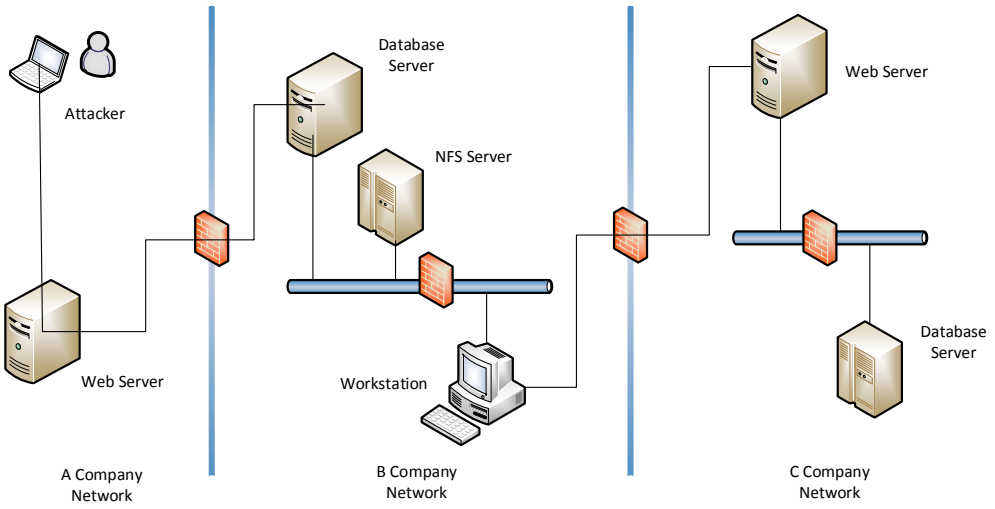


Figure 3: The Attack and Mission Scenario.

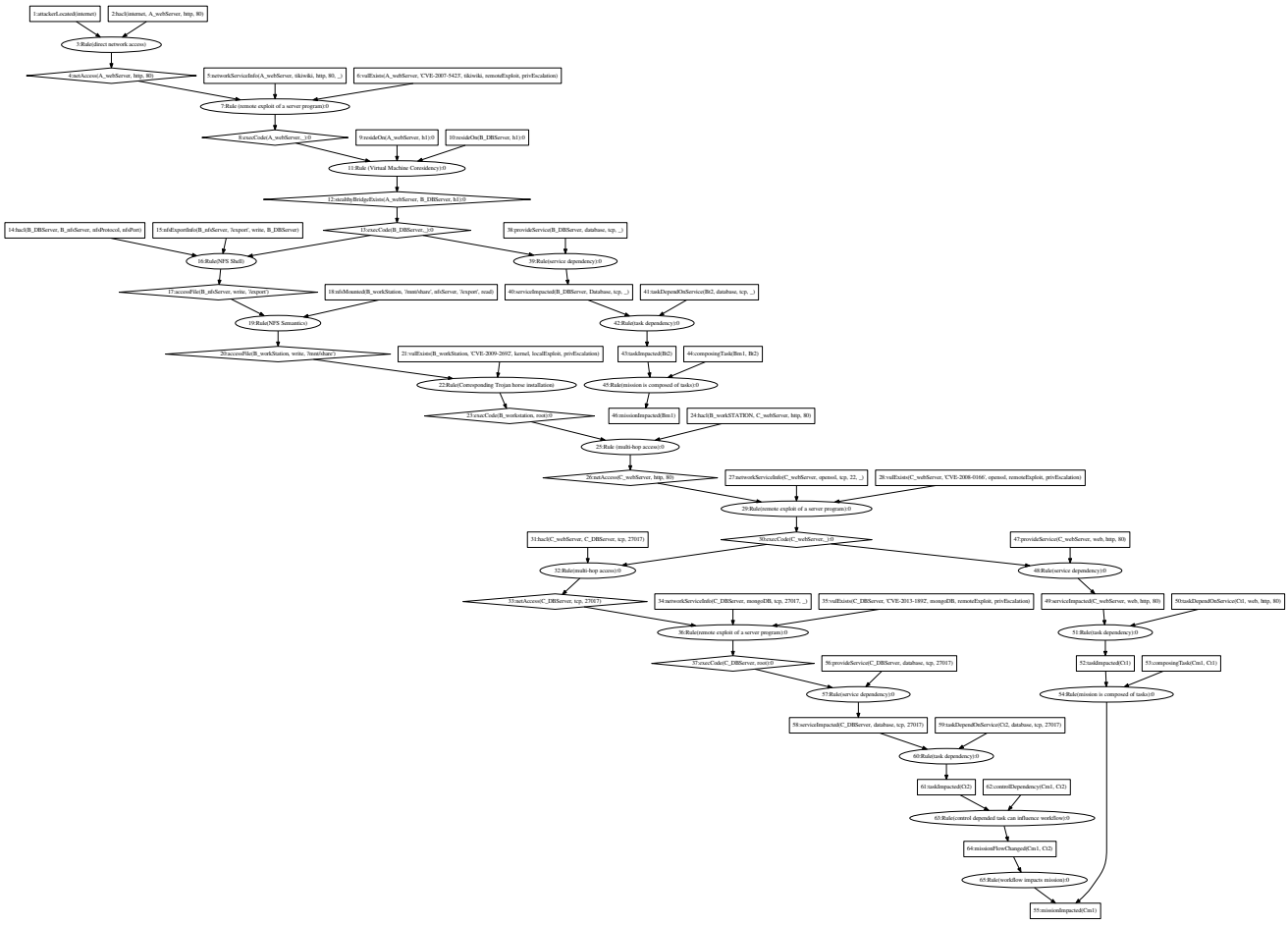


Figure 4: Mission Impact Graph.

Liu, Peng; Singhal, Anoop; Sun, Xiaoyan.
"Towards Actionable Mission Impact Assessment in the Context of Cloud Computing."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

input conditions. For example, what if we remove a server? What if we patch a vulnerability on a host? Which tasks or missions will be affected? In our case study, if we break the co-residency relationship between A's webserver and B's database server by moving one of the virtual machines, attacks towards B and C will prevented. As a result, missions in B and C won't be affected. Similarly, if the vulnerability on C's webserver is patched, attacks towards C can be stopped and mission *Cm2* will be safe. In addition, we can also analyze the potential mission impact by assuming vulnerability existence on other servers. For example, what if an unknown security hole exists on a host? Which tasks or missions will be affected in this case? In addition, as the situation knowledge regarding a network is continuously collected, such knowledge can be interpreted into input files to the automated tool for iterative "what-if" mission impact analysis based on the current situation. Therefore, performing such "what-if" analysis enables interactive mission impact analysis, and thus helps security admins make correct decisions for cyber resilience.

8. CONCLUSION

This paper makes the first efforts to close a gap between mission impact assessment and cyber resilience. In the cloud environment it is even more difficulty to analyze the impact of vulnerabilities and security events on mission. To fill the gap and associate missions with current attack-resilient systems, this paper develops a novel graphical model that interconnects mission dependency graphs and cloud-level attack graphs. Our case study shows that in some cases this model successfully bridges the gap and may significantly boost the cyber resilience of mission critical systems.

Disclaimer

This paper is not subject to copyright in the United States. Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

9. REFERENCES

[1] Proceedings of DARPA Information Survivability Conference and Exposition, Anaheim, California, 12-14 June 2001, Volume I & Volume II.

[2] A. D'Amico, L. Buchanan, J. Goodall. Mission Impact of Cyber Events: Scenarios and Ontology to Express the Relationships between Cyber Assets, Missions, and Users. Proc. 5th Int'l Conf. on Information Warfare and Security, 2010.

[3] J. Holsopple, S. J. Yang, and M. Sudit. Mission Impact Assessment for Cyber Warfare. Book chapter in Intelligent Methods for Cyber Warfare, Springer, 2015.

[4] Gabriel Jakobson. Mission Cyber Security Situation Assessment Using Impact Dependency Graphs. In Information Fusion (FUSION), 2011.

[5] R. Sawilla, X. Ou. Identifying critical attack assets in dependency attack graphs. Defense R&D Canada-Ottawa, Technical Memorandum, DRDC Ottawa TM 2008-180, 2008.

[6] S. Musman, A. Temin, M. Tanner, D. Fox, B. Pridemore. Evaluating the Impact of Cyber Attacks on Missions. MITRE Corporation, 2009.

[7] R. R. Yager. On ordered weighted averaging aggregation operation in multicriteria decision making. IEEE Transactions on Systems, Man and Cybernetics, Vol. 18, pages 183-190, 1988.

[8] X. Ou, W. F. Boyer, and M. A. McQueen. A scalable approach to attack graph generation. ACM CCS, 2006.

[9] X. Ou, S. Govindavajhala, and A. W. Appel. MulVAL: A Logic-based Network Security Analyzer. USENIX security, 2005.

[10] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs, in Security and Privacy (S&P), 2002.

[11] C. R. Ramakrishnan, R. Sekar, Model-based analysis of configuration vulnerabilities, Journal of Computer Security, 2002.

[12] C. Phillips and L. P. Swiler, A graph-based system for network-vulnerability analysis, in Proceedings of workshop on New security paradigms, 1998.

[13] S. Jajodia, S. Noel, and B. O'Berry, Topological analysis of network attack vulnerability, Managing Cyber Threats, 2005.

[14] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable graph-based network vulnerability analysis. ACM CCS, 2002.

[15] K. Ingols, R. Lippmann, and K. Piwowarski. Practical attack graph generation for network defense. ACSAC, 2006.

[16] Xiaoyan Sun, Jun Dai, Anoop Singhal, Peng Liu. Inferring the Stealthy Bridges between Enterprise Network Islands in Cloud Using Cross-Layer Bayesian Networks. SecureComm, 2014.

[17] Zhang, Yinqian, et al. Homealone: Co-residency detection in the cloud via side-channel analysis. 2011 IEEE Symposium on Security and Privacy. IEEE, 2011.

[18] Ristenpart, Thomas, et al. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[19] Younis, Younis, Kashif Kifayat, and Madjid Merabti. Cache side-channel attacks in cloud computing. International Conference on Cloud Security Management (ICCSM). 2014.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

A Secure Multicast Group Management and Key Distribution in IEEE 802.21

Yoshikazu Hanatani[1], Naoki Ogura[1], Yoshihiro Ohba[2], Lidong Chen[3], and Subir Das[4]

[1] Toshiba Corp., 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan.
{yoshikazu.hanatani, naoki.ogura}@toshiba.co.jp
[2] Toshiba Electronics Asia Pte.,Ltd., 20 Pasir Panjang Road, #12-25/28 Mapletree Business City, Singapore, 117439.
yoshihiro.ohba@toshiba.co.jp
[3] National Institute of Standards and Technology, 100 Bureau Dr. Gaithersburg, MD 20899-8930, U.S.A.
lily.chen@nist.gov
[4] Applied Communication Sciences, 150 Mount Airy Road, Basking Ridge, New Jersey, U.S.A., 07920.
sdas@appcomsci.com

Abstract. Controlling a large number of devices such as sensors and smart end points, is always a challenge where scalability and security are indispensable. This is even more important when it comes to periodic configuration updates to a large number of such devices belonging to one or more groups. One solution could be to take a group of devices as a unit of control and then manage them through a group communication mechanism. An obvious challenge to this approach is how to create such groups dynamically and manage them securely. Moreover, there needs to have mechanisms in place by which members of the group can be removed and added dynamically. In this paper, we propose a technique that has been recently standardized in IEEE 802.21 (IEEE 802.21-2015™) with the objective in providing a standard-based solution to the above challenges. The approach relies on Logical Key Hierarchy (LKH) based key distribution mechanism but optimizes the number of encryption and decryption by using "Complete Subtree". It leverages IEEE 802.21 framework, services, and protocol for communication and management. It provides a scalable and secure way to manage (e.g., add and remove) devices from one or more groups. We describe the group key distribution protocol in details and provide a security analysis of the scheme along with some performance results from a prototype implementation.

Keywords: Group Communication, group key and management, multicast, Group Key Block (GKB), Subtree, IEEE 802.21

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation, Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

1 Introduction

In today's networked world, it is becoming more and more expensive when it comes to configuring and software updates to large number of remote sensors and smart devices. To alleviate the cost and scalability issues, operators and vendors perform these operations remotely, commonly known as remote device management. While remote configurations updates are very common and secure networking technologies are available, normally it happens via a remote server in which each device requires to connect the server. This process becomes bandwidth inefficient (*n* unicast connections) and time consuming when the configuration update of a group of devices involves transferring a large amount of data. On the contrary, if these updates can be performed via a secure group communication mechanism whereby the network entity can multicast or broadcast the messages to a group of devices, the process becomes more efficient and saves a great deal of time and network resources,

IEEE 802.21-2008™ [1] defines a media independent framework, services and signaling protocol that are standardized in IEEE while the transport of the signaling protocol over IP is standardized in IETF. The standard published in [1] addresses the handover optimization use case whereby the user experience of ongoing application flows can be improved significantly for mobile nodes (MNs) that are moving from one link layer access technology to another irrespective of whether the access network is managed by the same or different network operators. The framework provides a signaling protocol that can be transported natively over the link layer or over Internet Protocol (IP) using underlying unicast and multicast mechanisms. In subsequent years of Standards amendment process, IEEE 802.21 Working Group addressed other use cases and defined signaling protocol and services security along with a group management mechanism in [2, 3]. In particular, Standards published in [3] targeted the use case where a large number of groups of devices are required to be managed from a group manager that resides in an entity in the network. Therefore [3] is relevant to our discussion in which a network entity can multicast a message to a group of nodes (or devices) using IEEE 802.21 media independent protocol interface, and secure group key distribution mechanism to cryptographically protect these multicast messages. The amendment [3] not only adds the secure group communication mechanism but also allows network nodes to communicate handover messages and to perform other management operations such as failover, failback, and configuration updates to a group of devices that are part of the network. The standardized approach relies on Logical Key Hierarchy (LKH) based key distribution mechanism [4, 5, 6] and uses "Complete Subtree" to optimize the number of encryption.

In this paper, we first introduce IEEE 802.21 [3] defined protocol and then discuss how to use the complete subtree method to optimize the performance of group communication. Subsequently, we introduce specific methods to handle the issues in group key distribution for IEEE 802.21 applications. In addition, we also analyze security of the group key distribution protocol as specified in IEEE 802.21 [3].

The paper is organized as follows: Section 2 discusses the related work. Section 3 presents the preliminaries of group key distribution approach while Section 4 describes the Group Key Block. Section 5 describes the group key distribution scheme and Section 6 provides a formal model-based security analysis. Section 7 captures our initial prototype implementation results and Section 8 concludes the paper.

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2 Related work and our approach

Secure multicast-based communication has been an important research topic in cryptography and in communication security. Most of the research discusses theoretical boundaries on the message length (i.e., number of encryptions), storage (i.e., number of keys each member holds), and computations for each receiver [7]. Some of the research also discusses trace-and-revoke algorithm with an upper bound of coalitions, which is outside the scope of this paper.

In practical applications, the secure group communications have been handled through initial pairwise group key distribution to group members [8, 9]. The schemes in [8] allow group key distribution for rekeying. On the other hand, whenever new members join the group or some current members leave the group, the schemes defined in [9] have to use pairwise secure channels for key distribution.

Logical Key Hierarchy (LKH) has been introduced in [4, 5, 6] for group key update, assuming each group member has been provisioned with one fixed individual key or the individual keys are established using other methods. The LKH is represented as a tree while the individual keys are represented as leafs of the tree. The nodes above the leaf level represent the keys shared by different members represented as leafs which have a path to the node. Every time, a member or members join or leave the group, the tree is updated.

The group key distribution scheme introduced in this paper uses a similar tree to represent the fixed keys that each group member hold. The group key is encrypted by a set of keys represented in the tree such that each member in the group owns a key to decrypt it, while the nodes not in the group do not have the proper decryption keys. Each time when group members join or leave, a new group key is distributed using the proper keys for the new group. Intuitively, in a given group, if more group members shared the same key, that is, their paths meet at the same node, the less encryptions are needed. In order to gain such efficiency, the scheme in this paper uses ‘Complete Subtree’. The ‘Complete Subtree’ method is introduced in [10] to optimize the number of encryptions and decryptions for each group key distribution. These methods have not been adopted in the practical applications to the best of our knowledge.

In particular, the method in this paper uses a single key tree to distribute keys for different groups. For a given group, our method generally requires less number of key encryptions than LKH for the key distributor, which also means lower transmission burden. Let L (> 1) denote the number of leaf nodes of the key tree, $N(< L)$ denote the number of root nodes of complete subtrees covering all leaf nodes of the members of the group, and M denote the number of ancestor nodes of the N root nodes. In initial group key distribution, LKH requires at least L encryptions to distribute the group key (which is the key corresponding to the root node of the key tree in LKH) and other keys to be used for key update. In group key update, LKH requires $(N + M)$ encryptions to update the group key for all group members excluding revoked members. In contrast, our method requires N encryptions of the group key for both initial group key distribution and group key update. Therefore, if we assume the same key tree size, our method

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

always requires less number of encryptions than LKH. Our method allows to take advantage of complete subtrees when possible. That is, when the group members represented by the leaf nodes can be grouped to complete subtrees, the number of key encryptions can be further reduced.

For any group member m, our method requires a single decryption to obtain the group key, while for LKH, $(H - H_m + 1)$ decryptions are needed where H represents the height of the key tree and H_m represents the height of the complete subtree that covers the leaf node of member m. In the applications where each group member is a constrained device such as a sensor, our method has significant advantages.

On the other hand, it has not been clear how scalable complete subtree is and no specific algorithms have been proposed to identify the complete subtrees. The use of a media independent framework and a signaling protocol that can be transported natively over Ethernet or over Internet Protocol (IP) using underlying unicast and multicast mechanisms is another important aspect that has not been standardized or published to the best of our knowledge.

While different security notions for group key agreement protocols have been introduced in [11, 12, 13, 14], we define a variant of another formal security model called Bresson and Manulis (BM) model [15] to satisfy the similar security requirements. The BM model cannot be applied directly for our security proof because the group key distribution protocol specified in [3] does not provide perfect forward secrecy.

3 Preliminaries

In this section, we introduce some basic concepts used in group management and key distribution. The concepts of key tree and complete subtree are essential for the key distribution protocol that we discuss in subsection 3.1. When a group key is distributed, it is protected by a key wrapping mechanism. To authenticate the sender, the encrypted group key is digitally signed. The security notions and definitions of key wrapping and signatures are introduced in subsections 3.2 and 3.3, respectively.

3.1 Key tree and complete subtree method

In this paper, we assume there are a large set of devices $\mathcal{U} = \{U_1, U_2, \ldots, U_m\}$ in which each device is provisioned with a set of keys, called *device keys*, DK_i. In the group key distribution protocol, the key is distributed by a group manager (GM) to a subset of the devices $S = \{U_{i1}, U_{i2}, \ldots, U_{ik}\}$.

A key tree is a binary tree with depth n and it has t levels from the root to the leafs. Therefore, such a key tree has 2^n leaf nodes whereby each leaf node is a device and to represent all the devices $\{U_1, U_2, \ldots, U_m\}$, it requires $m \leq 2^n$. **Figure 1** is an example of depth-3 tree. Each node (e.g., a leaf node, an inner node, or the root node) is coded with a binary string called *index* and a key. Assume the root node is on the top. The next level nodes have indices 0 and 1 from the left to the right. The corresponding

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

keys are denoted k_0 and k_1. The next level nodes have indices 00, 01 as decedents of node 0 while 10 and 11 as decedents of node 1. The corresponding keys are denoted k_{00}, k_{01}, k_{10}, k_{11}. Nodes in every level are indexed this way until the level t. In the rest of this paper, we will denote the node with the index and the key (I_i, k_i). We simply call each key as a node key labeled with its index. For the leaf node, we also use the integer converted from its index as the leaf number which maps to a specific device.

For device U_j, the provisioned device keys consist of all the node keys from the leaf along the path to the root. In the example depicted in **Figure 1**, a device represented by leaf "000" is provisioned with device keys {k(root), k_0, k_{00}, and k_{000}}.

In order to distribute a master group key to the devices in a specific group, the master group key **mgk** is protected with a set of keys in such a way that each device in the group must own a key in its device key set to recover the **mgk**, while for any device not in the group, it cannot recover the **mgk**. For a given group, there must be many different ways to protect the **mgk**. For example, in **Figure 1** consider a group represented by the leaf nodes 000, 001, 010, 011, 101, and 111. Notice that nodes 000 and 001 share the same key k_{00} and nodes 000, 001, 010, 011 all share the key k_0, then protection with the following key sets all satisfy the condition stated above.

a. k_{000}, k_{001}, k_{010}, k_{011}, k_{101}, k_{111};
b. k_{00}, k_{01}, k_{101}, k_{111};
c. k_0, k_{101}, k_{111}.

Obviously, key set c is more appealing because it calls the least number of the protection mechanisms and thus, generated the shortest of the ciphertext for broadcast. The concept of complete subtree is introduced to optimize the number of calls to the protection mechanisms. In this paper, the protection mechanisms can be a key wrapping algorithm or an encryption algorithm.

A complete (depth-l) subtree in a depth-t tree is a subtree with 2^l leaf nodes such that their indices have common prefix of t-l bits. For the tree in **Figure 1**, nodes represented with indices 000 and 001 form a depth-1 complete subtree at root 00, while nodes represented with indices 000, 001, 010, and 011 form a depth-2 complete subtree at root 0. For a subset of the group, if it can form a complete subtree, using the key represented by the subtree root allows all the members to recover the protected group key. Therefore, identifying complete subtrees in a given group can optimize the computation and communication resources in group key distribution for that group.

It shall be noticed that a single leaf is a depth-0 complete subtree. In fact, the optimization is to find out the non-overlapping maximum complete subtrees, which can cover the whole group. The set of non-overlapping maximum complete subtrees is unique for a given group. For example, for the group with the leaf nodes 000, 001, 010, 011, 101, 111, the set of the non-overlapping maximum complete subtrees that covers all the members is a depth-2 complete subtree and two depth-0 complete subtrees. Standards published in [3] specifies a complete subtree algorithm to determine such set.

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

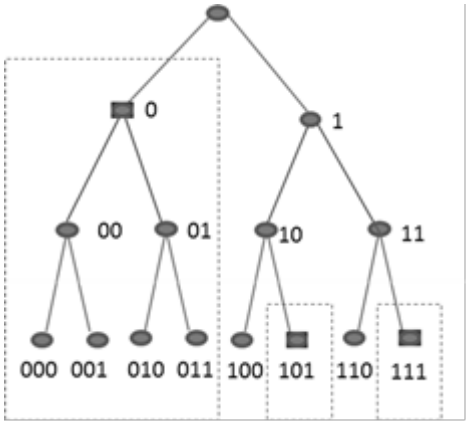


Figure 1. A depth-3 key tree

3.2 Key-wrapping scheme

After determining which keys to use through the complete subtrees, for the group key distribution, the key is protected by a key wrapping scheme. Key-wrapping scheme is a symmetric key encryption scheme for sending a group key. For group key distribution, IEEE std 802.21d-2015™ [3] supports two deterministic symmetric key schemes, AES-key-wrapping-128 and AES-ECB-128. Here $x \leftarrow\!\!\!\!- _R\ X$ means that x is an element chosen uniformly at random in a finite set X.

Definition 1. *Key-wrapping $\mathcal{KW}$ is a 3-tuple of algorithms* $(\mathsf{KeyGen}_{\mathcal{KW}}, \mathsf{Wrap}, \mathsf{Unwrap})$ *satisfying*:

- $\mathsf{KeyGen}_{\mathcal{KW}}$*: a probabilistic algorithm takes the security parameter* κ*, and returns* $K \in \{0,1\}^{\kappa}$,
- Wrap *: a deterministic algorithm takes* $K \in \{0,1\}^{\kappa}$ *and* $D \in \{0,1\}^{l}$*, and returns* $C \in C$ *where* l *is a bit-length of key to be wraped.*
- Unwrap *: a deterministic algorithm takes* $K \in \{0,1\}^{\kappa}$ *and* $C \in C$*, and returns* $D \in \{0,1\}^{l} \cup \{\perp\}$,
  where $\forall K \leftarrow \mathsf{KeyGen}_{\mathcal{KW}}(\kappa), \forall D \in \{0,1\}^{l}$: $\mathsf{Unwrap}(K, \mathsf{Wrap}(K,D))=D$.

A basic security requirement for symmetric encryption scheme is the indistinguishability against chosen plaintext attack (IND-CPA), and it is well-known that no deterministic encryption schemes can satisfy the IND-CPA. On the other hand, for sending a random key, the following weaker security requirement is sufficient.

Definition 2. *(Indistingutishability against Random-Plaintext Attack) Let* $b \leftarrow_R \{0,1\}$ *and* $W \leftarrow \mathsf{KeyGen}_{\mathcal{KW}}(\kappa)$ *where* κ *is a security parameter. The RPA-*

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

advantage of $\mathcal{A}$ can send queries to oracles Wrap_W *and* LR_W. *When the oracle* Wrap_W *receives a query,* Wrap_W *selects* $D \leftarrow_R \{0,1\}^l$, *and returns* $(D, \mathsf{Wrap}(W, D))$. *When the oracle* LR_W *receives a query,* LR_W *selects* $D_0, D_1 \leftarrow_R \{0,1\}^l$, *and returns* $(D_0, D_1, \mathsf{Wrap}(W, D_b))$. *The RPA-advantage of* $\mathcal{A}$ *is defined as*

$$\mathsf{Adv}^{\mathrm{kw.rpa}}_{\mathcal{A},\mathcal{KW}}(\kappa) = \Pr\left[\mathcal{A}^{\mathsf{Wrap}_W,\mathsf{LR}_W} \to 1 | b = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{Wrap}_W,\mathsf{LR}_W} \to 1 | b = 0\right]$$

where $\mathcal{A}$ *is a probabilistic polynomial-time algorithm that sends at most q queries to* Wrap_W *and at most 1 query to* LR_W.

$\mathcal{KW}$ *is IND-RPA secure if for all* $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{kw.rpa}}_{\mathcal{A},\mathcal{KW}}(\kappa)$ *is negligible.*

Random-Plaintext attack is originally defined in [16]. In the original definition in [16], the adversary is not allowed to query the Wrap_W. According to a similar discussion in [16], we can show that an ECB (electronic code book) mode based on a random permutation with block length *n* is IND-RPA secure per Definition 2.

3.3 Signature scheme

In order to authenticate a sender, IEEE std 802.21d-2015™ [3] supports one digital signature scheme, ECDSA (Elliptic Curve Digital Signature Algorithm). Here $x \longleftarrow_R X$ means that x is an element chosen uniformly at random in a finite set X.

Definition 3. *Signature* Σ *is a 3-tuple of algorithms* $(\mathsf{KeyGen}_{\mathcal{KW}}, \mathsf{Sign}, \mathsf{Verif})$ *satisfying:*

- KeyGen_Σ *: a probabilistic algorithm that takes the security parameter* κ, *and returns a pair of public key and secret key* (pk, sk),
- Sign *: a probabilistic algorithm takes sk and a message* $m \in \{0,1\}^*$, *and returns* σ,
- Verif *: a deterministic algorithm takes pk, m, and* σ, *and returns 0 or 1,*

where $\forall (pk, sk) \leftarrow \mathsf{KeyGen}_\Sigma(\kappa), \forall m \in \{0,1\}^*$ *:* $\mathsf{Verif}(pk, m, \mathrm{Sign}(sk, m)) = 1$.

Definition 4. *(Existential Unforgeability against Chosen Message Attacks) Let* $\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}, \mathsf{Verif})$ *be a digital signature scheme, and* $(pk, sk) \leftarrow \mathrm{KeyGen}_\Sigma(\kappa)$. *When a signing oracle* Sign_{sk} *receives a query* $m \in \{0,1\}^*$, *it returns* $\sigma = \mathrm{Sign}(sk, m)$. *The advantage of* $\mathcal{A}$ *is defined as*

$$\mathsf{Adv}^{\mathrm{euf-cma}}_{\mathcal{A},\Sigma}(\kappa) = \Pr[\mathcal{A}^{\mathsf{Sign}_{sk}}(pk) \to (m^*, \sigma^*) : \mathsf{Verif}(pk, m^*, \sigma^*) = 1 \wedge m^* \notin \mathcal{M}]$$

where $\mathcal{M}$ *is the set of message queried to* Sign_{sk} *and* $\mathcal{A}$ *is a probabilistic polynomial-time algorithm who sends at most* q_S *queries to* Sign_{sk}.

Σ *is EUF-CMA secure if for all* $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{euf-cma}}_{\mathcal{A},\Sigma}(\kappa)$ *is negligible.*

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

4 Group Key Block

Group Key Block (GKB) is a data format defined in IEEE std 802.21d-2015™ [3] for encoding a group key and other data associated with the group key. The following attributes are contained in a GKB:

- GroupKeyData: a list of octet strings, each of them contains the group key encrypted by using a distinct node key specified in 'CompleteSubtree'. Either AES_Key_Wrapping-128 or AES_ECB-128 is used for encrypting the group key.
- GroupIdentifier: an identifier of a group.
- CompleteSubtree: a list of node indices corresponding to root nodes of specific subtrees of the key management tree. See Section 4.1 for more details.
- SubgroupRange: a range of valid leaf identifiers in the 'CompleteSubtree'. A 'SubgroupRange' is used when a GKB is fragmented into multiple smaller pieces (see Section4.2).
- VerifyGroupCode: a pre-known octet string encrypted by the group key. A 'VerifyGroupCode' is used for checking whether the decrypted group key is the same as the one generated by the GM. 'VerifyGroupCode' may be used when AES_ECB-128 is used for group key encryption (Note that AES_Key_Wrapping has a built-in key verification mechanism).

Digital signature is added to each message carrying a GKB using the signature scheme described in Section 3.3.

4.1 Encoding complete subtrees

IEEE std 802.21d-2015™ [3] defines three methods for encoding 'CompleteSubtree'. In this section, the default encoding method is explained. In the default encoding method, a list of the node indices is contained in the 'CompleteSubtree' where each node index in the key management tree represents the root node of a distinct subtree covering only leaf nodes corresponding to members of the group. A node index consists of a depth in the key management tree and a subindex that is unique within the depth. In 'GroupKeyData', i-th string contains the group key encrypted by the node key corresponding to the i-th node index in the 'CompleteSubtree'.

4.2 GKB fragmentation

As described in Section 4.1, the size of CompleteSubtree in Method 1 is proportional to the number of subtrees encoded. Also, when a GKB is multicast and the number of recipient is large (e.g., thousands or more), it is difficult to reliably deliver the GKB to all recipients.

IEEE std 802.21d-2015™ [3] addresses this issue by defining a special fragmentation mechanism for fragmenting GKB. Unlike other general-purpose fragmentation mechanisms (e.g., IP fragmentation), a recipient of GKB does not have to receive all the frag-

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

ments of a single complete GKB and reassemble into the original GKB data. The recipient can instead determine whether it is a member of the group and can obtain the group key by receiving one GKB fragment that contains in 'SubgroupRange' attribute.. Suppose a single complete GKB is fragmented into five GKB fragments with 'SubgroupRange' of each fragment set to (0,99), (100,199), (200,299), (300,399), and (400,499). A recipient whose leaf identifier is 250, when receiving the GKB fragment with 'SubgroupRange (200,299)', can determine whether it is a member of the group. Thus it can obtain the group key if it is a group member and can simply ignore other four GKB fragments.

5 Group key distribution protocol

IEEE std 802.21d-2015™ [3] defines an architecture and a group key distribution protocol that a group manager (GM) can use to communicate to group members via a multicast transport. The group key distribution protocol uses the 'Complete Subtree' method with a deterministic symmetric key encryption scheme and a digital signature scheme. In this section, we introduce a simplified version of the group key distribution protocol using an option that is described in [3]. In this section we refer a group member to a user.

Provisioning

IEEE std 802.21d-2015™ [3] assumes that a group manager and each user has device keys, which are also called long-term keys. The secure provisioning method is not defined in the standard.

1. Let 2^n be the number of (potential) users managed by the group manager GM, and let $\mathcal{U}$ be a set of all users. GM generates a key tree T with depth n, and assigns (I_i, k_i) to each node in T where I_i is a node index represented as a binary string of length between 1 to n and $k_i \leftarrow \mathsf{KeyGen}_{\mathcal{KW}}(\kappa)$ is a node key where i corresponding to the node index I_i. For digital signature, GM generates $(pk, sk) \leftarrow \mathsf{KeyGen}_{\Sigma}(\kappa)$.
2. For all user U_i in $\mathcal{U}$, GM assigns each user U_i to a leaf node in T. Let $Path_{U_i}$ be a set of node indices of nodes from the leaf node which is assigned to U_i along the path to the root node. GM assigns $DK_i = \{(I_j, k_j)\}_{I_j \in Path_{U_i}}$, to U_i as the long-term keys.
3. GM securely sends *pk* and DK_i to each of U_i.

Procedure of GM

1. Decide a set of group members S which is a target for group key distribution and a group identifier GI which identifies a group using the distributed group key.
2. Pick a current sequence number *SN* for GI.

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3. Decide a destination group DG for the group key distribution message. GM is required to send the group key distribution message to all of its members S. For simplicity, we assume that DG includes S. A broadcast group BG including all users may be used as DG.
4. Select a master group key $\mathsf{mgk} \in \{0,1\}^l$ uniformly at random and select a security association identifier SAID which is an identifier of a group session key $\mathrm{gsk} = \mathrm{KDF}(\mathsf{mgk})$ where KDF is a key derivation function which is publicly shared.
5. Compute a list of indices CS from $\mathcal{U} \backslash S$ and T by 'Complete Subtree' method.
6. For all $I_i \in CS$, compute $c_i = \mathsf{Wrap}(k_i, \mathsf{mgk})$ where (I_i, k_i) is a node of T, and adds c_i to a group key data $GKD = GKD||c_i$.
7. Read a sequence number sq for the destination group DG.
8. Compute $\sigma = \mathrm{Sign}(sk, GI||SN||CS||GKD||\mathsf{SAID}||sq)$.
9. Send $(GI||SN||CS||GKD||\mathsf{SAID}||sq||\sigma)$ to DG.

Procedure of receiver U_i

1. Receive $(GI||SN||CS||GKD||\mathsf{SAID}||sq||\sigma)$.
2. Check sq whether the received message is not a replay attack. If the message with sq was already accepted, U_i stops the subsequent procedure.
3. If $\mathsf{Verif}(pk, GI||SN||CS||GKD||\mathsf{SAID}||sq, \sigma) \neq 1$, U_i stops the subsequent procedure.
4. If U_i has $(I_j, k_j) \in DK_i$ such that $I_j \in CS$,
   (a) compute $\mathsf{mgk} = \mathsf{Unwrap}(k, c_k)$ where $c_k \in GKD$ is the ciphertext corresponding with I_j,
   (b) compute the group session key $\mathrm{gsk} = \mathrm{KDF}(\mathsf{mgk})$, and record $(GI, SN, \mathsf{SAID}, \mathrm{gsk})$.

6 Security Analysis

6.1 Security requirements

We define a formal security model based on BM model [15]. Our security model modifies the definition of *freshness* in BM model to remove *perfect forward secrecy*. This is due to the reason that for IEEE 802.21 applications, reducing the number of multicast communication traffic is an important requirement.

Attack model. Let an adversary $\mathcal{A}$ and the users (including the group manager GM) be probabilistic polynomial-time algorithms. In order to capture multiple sessions, each user U is modeled by an oracle Π_U^s for $s \in \mathbb{N}$. Every session is identified by a unique, publicly-known sid_U^s. Let pid_U^s be a partner id that contains the identities of participating users (including U), and $\mathcal{G}(\Pi_{U_j}^s) = \{\Pi_{U_j}^t \text{where } U_j \in \mathsf{pid}_{U_i}^s \text{and } \mathsf{sid}_{U_i}^s = \mathsf{sid}_{U_j}^t\}$. $\Pi_{U_i}^s$ and $\Pi_{U_j}^t$ are called partner if $\Pi_{U_j}^t \in \mathcal{G}(\Pi_{U_i}^s)$ and $\Pi_{U_i}^s \in \mathcal{G}(\Pi_{U_j}^t)$. $\mathcal{A}$ learns each message

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

to be sent, and it can prevent sending or modifying the message. We assume that receivers always receive the original message sent by the sender, even if $\mathcal{A}$ blocks or modifies it.

$\mathcal{A}$ issues following queries.

- *Initialize*(S): For each user in the set S, a new oracle Π_U^s is initialized and the resulting session id sid is given to $\mathcal{A}$.
- *Invoke*(sid, S'): It assumes that sid is a valid session id and S' is a set of initialized oracles ($S' \subset S$ where S led to the construction of sid). In response, for each $U \in S'$, the oracle Π_U^s turns into the processing stage. If Π_U^s is an initiator of the protocol, Π_U^s outputs the first protocol message.
- *Send*(Π_U^s, m): The message m is sent to Π_U^s. In response, $\mathcal{A}$ receives a processing result of m based on the protocol. The response may be empty, if m is incorrect.
- *Corrupt*(U): In response, $\mathcal{A}$ obtains the long-term key of U, LL_U.
- *AddUser*(U, Λ): In response, a new user U with a long-term key is added to $\mathcal{U}$ where Λ contains the registration information and the long-term key. If the protocol prohibits U from selecting the long-term key, the long-term key in Λ is empty, and in response $\mathcal{A}$ additionally receives U's long-term key.
- *RevealState*(Π_U^s): In response, $\mathcal{A}$ obtains ephemeral secrets stored in state_U^s.
- *RevealKey*(Π_U^s): In response, $\mathcal{A}$ obtains the group session key k_U^s (only if Π_U^s has already accepted).

We say U is corrupted if LL_U is known to $\mathcal{A}$, either via *Corrupt*(U) or *AddUser*(U, Λ); if no such queries have been asked then U is honest.

Definition 5. *(Oracle Freshness) In a session sid of P, an oracle Π_U^s has accepted is fresh if all of the following holds:*

1. *no $U' \in \mathsf{pid}_U^s$ has been added by $\mathcal{A}$ via corresponding AddUser query,*
2. *no $U' \in \mathsf{pid}_U^s$ has been corrupted via corresponding Corrupt query,*
3. *neither Π_U^s nor any of its partners is asked for a query RevealState until Π_U^s and its partners accept,*
4. *neither Π_U^s nor any of its partners is asked for a query RevealKey after having accepted.*

In the original definition in [2], the condition 2. is "*no $U' \in \mathsf{pid}_U^s$ is asked for a query Corrupt prior to a query of the form Send($\Pi_{U_j}^t$, m) with $U_j \in \mathsf{pid}_U^s$ until Π_U^s and its partners accept*". It means that Π_U^s is *fresh* even if $U' \in \mathsf{pid}_U^s$ who is a participant of a future session is corrupted, i.e., it represents *perfect forward secrecy*.

In order to provide a formal security proof of our protocol without *perfect forward secrecy*, we modify condition 2 for the weaken $\mathcal{A}$ as descried in Def. 5.

Definition 6. *(Authenticated Key Exchange (AKE) security) Let P be a group key distribution protocol. Let $b \leftarrow_R \{0,1\}$ and $\mathcal{A}$ be an adversary against AKE security of P. The attack game $\mathsf{Game}_{\mathcal{A},\mathsf{P}}^{\mathrm{ake}-b}$ is defined as follows.*

1. *$\mathcal{A}$ interacts with each oracle using the queries defined in section 6.1.*

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2. *$\mathcal{A}$ sends Test query to Π_U^s in arbitrary timing. Π_U^s returns k_U^s if $b = 0$, else U returns a key k_r chosen from the key space uniformly at random.*
3. *$\mathcal{A}$ continues to interact with each oracles using the queries defined in section 6.1.*
4. *$\mathcal{A}$ outputs $b' \in \{0,1\}$ and stops.*

Let Fr *be an event that Π_U^s who receives Test query is still Fresh when $\mathcal{A}$ has been stopped. The advantage of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}_{\mathcal{A},\mathsf{P}}^{\mathrm{ake}-b}(\kappa) = |\Pr\left[\mathsf{Game}_{\mathcal{A},\mathsf{P}}^{\mathrm{ake}-b}(\kappa) = \mathrm{b} \wedge \mathsf{Fr}\right] - \frac{1}{2}|.$$

We say that a protocol P *is AKE secure if for all probabilistic polynomial-time adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A},\mathsf{P}}^{\mathrm{ake}-b}(\kappa)$ is negligible.*

6.2 Security proof

Theorem 1. *If Σ satisfies EUF-CMA security and $\mathcal{KW}$ satisfies IND-RPA security, the protocol* P *described in section 6 satisfies AKE-security in Def. 6, and*

$$\mathsf{Adv}_{\mathcal{A}_{AKE},\mathsf{P}}^{\mathrm{gk}-b}(\kappa) \leq \frac{(2N-1)\cdot n_s \cdot n_g^*}{2} \cdot \mathsf{Adv}_{\mathcal{A},\,\mathcal{KW}}^{\mathrm{kw.rpa}}(\kappa) + \mathsf{Adv}_{\mathcal{A},\Sigma}^{\mathrm{euf}-cma}(\kappa)$$

where N is the maximum number of users, n_g^* is the number of ciphertexts containing within GKD in $\Pi_{U^*}^s$ who is the receiver of Test query, and n_s is the maximum number of sessions.

Proof of Theorem 1. The security proof is given by the game hopping technique [18]. Let S_i be an event that $b = b'$ and Π_U^s who receives Test query is *fresh* at the end of Game *i*.

Game 0: The original attack game of AKE security. Due to Def. 6,

$$\mathsf{Adv}_{\mathcal{A}_{AKE},\mathsf{P}}^{\mathrm{gk}-b}(\kappa) = |\Pr[S_0] - 1/2|. \quad (1)$$

Game 1: Let $\mathrm{L_M}$ be a list of messages issued by GM. In Game 1, each Π_U^s ignores $Send(\Pi_U^s, m)$ if $m \notin \mathrm{L_M}$, and other operations are the same as Game 0.

In the protocol P, the protocol message without the valid signature of GM is dropped by the receivers. The behavior of Π_U^s may be different between Game 0 and Game 1, if and only if $\mathcal{A}_{\mathrm{AKE}}$ succeeds the existential forgery of Σ. We assume Σ is EUF-CMA secure, then

$$|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Adv}_{\mathcal{A},\Sigma}^{\mathrm{euf}-cma}(\kappa). \quad (2)$$

Game 2: Let L_U be a list of messages received by the user U. In Game 2, Π_U^s ignores $Send(\Pi_U^s, m)$ if $m \in \mathrm{L}_U$, and other operations are the same as Game 1.

The message of P contains the sequence number *sq*, and each receiver does not accept the same sequence number. The number of Sent queries is at most polynomial times in

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

κ since $\mathcal{A}$ is a polynomial-time algorithm. If the space of *sq* is exponentially large in κ, the behaviors of Π_U^s in Game 1 and Game 2 are the same, then

$$\Pr[S_2] = \Pr[S_1]. \tag{3}$$

Game 3: GM try to guess a session s^* that $\mathcal{A}_{\mathrm{AKE}}$ sends Test query and (I^*, k^*) used in s^*. If it finds that the guess is failed, Game 3 is aborted, and GM decides b' which is the output of Game 3 instead of $\mathcal{A}_{\mathrm{AKE}}$. Game 3 is the same as Game 2 excluding following operations;

- After the *Setup* phase, GM randomly selects a session $s^* \in \{1, \ldots, n_s\}$ and a node in the key management tree *T* which has *N* leaf nodes[1], for guessing the session s^* and the node key k^* used in s^*. Let *Hit* be an event that GM succeeds the guess[2].
- When it finds *Hit* does not occur, Game 3 is aborted and GM decides $b' \leftarrow_{\mathrm{R}} \{0,1\}$ instead of $\mathcal{A}_{\mathrm{AKE}}$.

When *Hit* occurs, Game 3 and Game 2 are the same. When *Hit* does not occur, S_3 occurs at random since GM selects $b' \leftarrow_{\mathrm{R}} \{0,1\}$.

$$\begin{aligned}\Pr[S_3] &= \Pr[S_3 \wedge Hit] + \Pr[S_3 \wedge \neg Hit] \\ &= \Pr[Hit]\Pr[S_3|Hit] + \Pr[\neg Hit]\Pr[S_3|\neg Hit] \\ &\geq \frac{1}{(2N-1)n_s}\Pr[S_2] + \frac{1}{2} - \frac{1}{2(2N-1)n_s}\end{aligned} \tag{4}$$

where *N* is the maximum number of users, n_s is the maximum number of the sessions.

Game 4: In order to estimate $|\Pr[S_3] - 1/2|$, we consider the following hybrid game. In order to replace the reply of Test query with C where $(D_0, D_1, C) \leftarrow \mathrm{LR}_W$, a node key k^* is replaced by *W* using an Wrap_W oracle. Game 4 is the same as Game 3 excluding the following operations;

- For a session s excluding s^*, when the group manager Π_{GM}^s needs a ciphertext with k^* for generating the encrypted group keys GKD, Π_{GM}^s accesses Wrap_W oracle and it receives (D, C). D is regarded as a master group key mgk distributed in s, GKD is generated by C and D with other node keys excluding k^* in *T*, e.g., $\mathrm{Wrap}(k_{i_1}, D), \ldots, \mathrm{Wrap}(k_{i_{k-1}}, D), C, \mathrm{Wrap}(k_{i_{k+1}}, D), \ldots, \mathrm{Wrap}\left(k_{k_{n_k}}, D\right)$ where n_k is the number of ciphertext contained in GKD.
- For the session s^*, the group manager $\Pi_{GM}^{s^*}$ sends a query to LR_W and receives (D_0, D_1, C^*). D_0 is regarded as a master group key mgk distributed in s^*, and D_1 is regarded as a random key. GKD^* is generated by C^*, D_0, and D_1 with other node keys excluding k^*in *T*, e.g.,

[1] The complete binary tree *T* with *N* leaf nodes has (2*N*-1) nodes.

[2] If the guess is correct, i.e., *Hit* occurs, no U^* assigned (I^*, dk^*) is corrupted at the end of Game 3 since Π_U^* who receives Test query must be *fresh.*

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

$$\mathrm{Wrap}(k_{i_1}, D_0), \ldots, \mathrm{Wrap}\left(k_{i_{j-1}}, D_0\right), C^*, \mathrm{Wrap}\left(k_{i_{j+1}}, D_1\right), \ldots, \mathrm{Wrap}\left(k_{i_{n_g^*}}, D_1\right)$$

where n_g^* is the number of ciphertext contained in GKD^*

— When $\Pi_{U_j}^{s^*}$ receives Test query, it returns a group session key $\mathrm{KDF}(D_0)$.

Let $\mathrm{H_j}$ be a hybrid game that $GKD^* = \mathrm{Wrap}(k_{i_1}, D_0), \ldots, \mathrm{Wrap}\left(k_{i_{j-1}}, D_0\right)$, $\mathrm{Wrap}\left(k_{i_j}, D_0\right), \mathrm{Wrap}\left(k_{i_{j+1}}, D_1\right), \ldots, \mathrm{Wrap}\left(k_{i_{n_g^*}}, D_1\right)$. Let b_r be the bit of *IND-RPA* game. If $b_r = 0$, Game 3 is the same as $H_{\mathrm{j+1}}$ since C^*= $\mathrm{Wrap}(W, D_0)$. If $b_r = 1$, Game 3 is the same as H_j since C^*= $\mathrm{Wrap}(W, D_1)$. Let E_i be an event that occurs if $\mathcal{A}_{\mathrm{AKE}}$ outputs 1 in H_{i}, then $|\Pr[E_{i-1}] - \Pr[E_i]| \leq \mathsf{Adv}^{\mathrm{kw.rpa}}_{\mathcal{A},\,\mathcal{KW}}(\kappa)$ holds. By the hybrid argument, we have

$$\left|\Pr[E_0] - \Pr\left[E_{n_g^*}\right]\right| = \sum_{i=1}^{n_{g^*}} |\Pr[E_{i-1}] - \Pr[E_i]| \leq n_g^* \cdot \mathsf{Adv}^{\mathrm{kw.rpa}}_{\mathcal{A},\,\mathcal{KW}}(\kappa).$$

Accordingly, H_0 is the same as Game 3 when $b = 1$, and $H_{n_g^*}$ is the same as Game 3 when $b = 0$;

$$\begin{aligned}
\Pr[\mathrm{S_3}] - 1/2 &= |\Pr[\mathcal{A}_{\mathrm{AKE}} \to 1 \text{ in Game 3 } \wedge \mathrm{b} = 1] \\
&\quad + \Pr[\mathcal{A}_{\mathrm{AKE}} \to 0 \text{ in Game 3 } \wedge \mathrm{b} = 0] - 1/2| \\
&= \frac{1}{2}|\Pr[\mathcal{A}_{\mathrm{AKE}} \to 1 \text{ in Game 3 } |\mathrm{b} = 1] \\
&\quad - (1 - \Pr[\mathcal{A}_{\mathrm{AKE}} \to 1 \text{ in Game 3 } |\mathrm{b} = 0]) - 1| \\
&= \tfrac{1}{2}\left|\Pr[E_0] - \Pr\left[E_{n_g^*}\right].\right|
\end{aligned} \tag{5}$$

According to Eq. (1), (2), (3), (4), and (5),

$$\begin{aligned}
&\mathsf{Adv}^{\mathrm{gk}-b}_{\mathcal{A}_{AKE},\mathrm{P}}(\kappa) = \left|\Pr[S_0] - \frac{1}{2}\right| \\
&= \left|\Pr[\mathrm{S_1}] + |\Pr[\mathrm{S_1}] - \Pr[\mathrm{S_0}]| - \frac{1}{2}\right| \leq \left|\Pr[S_1] + \mathsf{Adv}^{\mathrm{euf}-cma}_{\mathcal{A},\Sigma}(\kappa) - \frac{1}{2}\right| \\
&= \left|\Pr[\mathrm{S_2}] + \mathsf{Adv}^{\mathrm{euf}-cma}_{\mathcal{A},\Sigma}(\kappa) - \frac{1}{2}\right| \\
&= \left|(2N-1) \cdot n_{\mathrm{s}}\left(\Pr[\mathrm{S_3}] - \frac{1}{2}\right) + \mathsf{Adv}^{\mathrm{euf}-cma}_{\mathcal{A},\Sigma}(\kappa)\right| \\
&\leq \frac{(2N-1) \cdot \mathrm{n_s} \cdot \mathrm{n_g^*}}{2} \cdot \mathsf{Adv}^{\mathrm{kw.rpa}}_{\mathcal{A},\,\mathcal{KW}}(\kappa) + \mathsf{Adv}^{\mathrm{euf}-cma}_{\mathcal{A},\Sigma}(\kappa).
\end{aligned}$$

7 Prototyping

We implemented the group key distribution protocol as described in section 5 and measured the processing time of group manager (GM) and receivers. In real systems, the receivers may have memory constraints. For such system, the code footprint size is also important. Therefore, we also measure the footprint size of the receivers. Table 1 shows the benchmark of the computing machine used for GM and receivers.

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Table 1. Computing Machine Specification

	GM	Receivers
CPU	Core i5-4310M, 2.7 GHz	ARM11176JZF-S, 700 MHz
RAM	4GB	512 MB
OS	Ubuntu 14.04.4	Raspbian

We considered the number of receivers is 1024, with threshold of fragmentation as 32. Table 2 shows processing time that GM takes to generate the protocol messages and the receiver takes to process them. During processing cycle, signing time and verification time of ECDSA are dominant, when ECDSA is attached to each messages. Table 3 shows the size of the protocol messages. The processing time and message size depend on the selection of group members.

Table 2. Processing times

	GM		Receiver	
	Average [msec]	Max [msec]	Average [msec]	Max [msec]
Best case	4.71	4.74	265.15	303.59
Worst case	83.80	85.01	4253.91	4276.05

Table 3. Protocol message size

	Message size[bytes]
Best case	272
Worst case	18336

In a best case scenario, $GroupKeyData$ contains only one ciphertext and hence GM sends only one message. In worst case scenario, GM sends 512 ciphertexts, where GM issues 16 messages with $GroupKeyData$ which contains 32 ciphertexts (given the threshold of fragmentation is 32). In our implementation, when the receiver receives a message, it first verifies an ECDSA signature in the message. Therefore in worst case scenario, receiver verifies 16 messages and hence the processing time increases. On the decryption side though the receiver needs to decrypt only one message that carries the complete subtree covering the receiver in order to extract the group key, and other 15 messages can be ignored after verification. So even in worst case, there is a significant advantage in terms of overall processing time.

Table 4 shows foot print size of the receivers.

Table 4. Footprint size of Receivers

	size [Byte]
heap	55264
stack	12200
text	172257
data	1268
bss	119900

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

Total	360889

We measured the memory usage occupied on the virtual memory space. The *text* segment corresponds to the code size. The *data* and *bss* segments include pre-defined variables (*data* has initialized data, while *bss* is uninitialized). The *stack* area is used for storing temporal variables on the program that is executed. The *heap* area is managed by malloc()-like functions. Each size of *text*, *data*, and *b*ss segments are fixed in every program executions. These values are measured by the size of the command. In this early prototype implementation, we focus on reducing the heap size for to simplify memory management simplification. The maximum sizes of *stack* or *heap* areas are measured by valgrind™ [19].

8 Conclusion

We introduced a secure multicast-based group key management and key distribution protocol that is recently standardized in IEEE 802.21. Although it is based on the concept of logical key hierarchy, a method has been specified on how the 'Complete Subtree' can be used to optimize the number of encryptions for each group key distribution. A data format called 'Group Data Block' has been used for encoding the 'Complete Subtree' and other data associated with it. To support the practical applications, the standard assumes an architecture whereby a group manager is responsible for distributing the group key. The group key distribution protocol uses a deterministic symmetric key wrapping scheme and a digital signature scheme. A formal security analysis and corresponding proof have been performed based on Bresson and Manulis model. While additional work is required, an early prototype implementation results with 1024 nodes and tree depth of 7 show that the scheme is realizable in memory constrained devices. It provides an easy way to securely add and remove group members.

References

1. IEEE Standard for Local and metropolitan area networks- Part 21: Media Independent Handover Services- IEEE Std 802.21™-2008, January 2009.
2. IEEE Standard for Local and metropolitan area networks- Part 21: Media Independent Handover; Amendment 1: Security Extensions to Media Independent Handover Services and Protocol, May 2012.
3. IEEE Standard for Local and metropolitan area networks- Part 21: Media Independent Handover; Amendment 4: Multicast Group Management, July 2015.
4. D. Wallner, E. Harder, and R. Agee Key Management for Multicast: Issues and Architectures Request for Comments: 2627 June 1999.
5. C. K. Wong, M. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," IEEE/ACM Transactions on Networking, vol. 8, No. 1, pp. 16-30, 2000.
6. ISO/IEC 11770-5 Information technology – Security techniques - Key management – Part 5: Group key management, 2011.
7. A. Fiat and M. Naor, Broadcast Encryption, Advances in Cryptology - CRYPTO'93, LNCS 773, Springer, 1994, pp. 480-491.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

8. B. Weis, S. Rowles, and T. Hardjono The Group Domain of Interpretation IETF, Request for Comments 6407, October 2011
9. IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2015.
10. D. Naor, M. Naor, and J. Lotspiech, Revocation and tracing schemes for stateless receivers. In Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings, pages 41–62, 2001.
11. Diffie, P.C. van Oorschot, M.J. Wiener, Authentication and authenticated key exchanges. Des. Codes Cryptography, 2(2):107–125, 1992.
12. M. Burmester, On the risk of opening distributed keys. In Advances in Cryptology CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings, pages 308–317, 1994
13. Y. Kim, A. Perrig, and G. Tsudik, Simple and fault-tolerant key agreement for dynamic collaborative groups. In CCS 2000, Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece, November 1-4, 2000., pages 235–244, 2000.
14. C. G. Gu¨nther. An identity-based key-exchange protocol. In Advances in Cryptology EUROCRYPT '89, Workshop on the Theory and Application of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings, pages 29–37, 1989.
15. T. Brecher, E. Bresson, and M. Manulis, Fully robust tree-diffie-hellman group key exchange. In Cryptology and Network Security, 8th International Conference, CANS 2009, Kanazawa, Japan, December 12-14, 2009. Proceedings, pages 478–497, 2009.
16. R. Gennaro and S. Halevi, More on key wrapping. In Selected Areas in Cryptography 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers, pages 53–70, 2009.
17. Burton H. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM, vol. 13, Issue 7, July 1970.
18. V. Shoup, Sequences of games: a tool for taming complexity in security proofs, IACR Cryptology ePrint Archive, 2004:332, 2004.
19. Valgrind, http://valgrind.org/

Chen, Lidong; Das, Subir; Hanatani, Yoshikazu; Ogura, Naoki; Ohba, Yoshihiro.
"A Secure Multicast Group Management and Key Distribution in IEEE 802.21."
Paper presented at 3rd International Conference on Research in Security Standardisation,
Gaithersburg, MD. December 5, 2016 - December 6, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Threat Modeling for Cloud Data Center Infrastructures

Nawaf Alhebaishi[1,2], Lingyu Wang[1], Sushil Jajodia[3], and Anoop Singhal[4]

[1] Concordia Institute for Information Systems Engineering, Concordia University
[2] Faculty of Computing and Information Technology, King Abdulaziz University
{n_alheb,wang}@ciise.concordia.ca
[3] Center for Secure Information Systems, George Mason University
jajodia@gmu.edu
[4] Computer Security Division, National Institute of Standards and Technology
anoop.singhal@nist.gov

Abstract. Cloud computing has undergone rapid expansion throughout the last decade. Many companies and organizations have made the transition from traditional data centers to the cloud due to its flexibility and lower cost. However, traditional data centers are still being relied upon by those who are less certain about the security of cloud. This problem is highlighted by the fact that there only exist limited efforts on threat modeling for cloud data centers. In this paper, we conduct comprehensive threat modeling exercises based on two representative cloud infrastructures using several popular threat modeling methods, including attack surface, attack trees, attack graphs, and security metrics based on attack trees and attack graphs, respectively. Those threat modeling efforts provide cloud providers practical lessons and means toward better evaluating, understanding, and improving their cloud infrastructures. Our results may also imbed more confidence in potential cloud tenants by providing them a clearer picture about potential threats in cloud infrastructures and corresponding solutions.

1 Introduction

Cloud computing has emerged as an attractive option for many enterprises, government agencies and organizations due to its flexibility and reduced costs. The shifting to this new paradigm is, however, still impeded by various security concerns, which are exacerbated by the lack of a clear understanding of security threats facing cloud data centers. Unlike traditional computer networks, cloud data centers usually exhibit some unique characteristics, such as the presence of significant redundancy in terms of hardware configurations, and the co-existence of both physical and virtual components. Such unique characteristics imply the need for modeling and measuring security threats specifically for cloud data centers.

On the other hand, modeling and measuring security threats for cloud data centers is a challenging task due to the lack of public accesses to the detailed information regarding hardware and software configurations deployed in real cloud data centers. Existing work mainly focus on high level frameworks for risk and impact assessment [19], guidelines or frameworks for cloud security metrics [2, 14], and specific vulnerabilities or threats in the cloud [6, 21] (a more detailed review of related work will be given in Section 6). However, to the best of our knowledge, there lack a concrete study on threat

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

modeling and measuring for cloud data centers using realistic cloud infrastructures and well established models. Although there already exist many such threat modeling models, such as attack surface, attack tree, attack graph, and their corresponding security metrics, a systematic application of those models to concrete cloud data center infrastructures is yet to be seen.

In this paper, we present a comprehensive study on modeling and measuring threats in cloud data center infrastructures. We first provide the basis for our study as two representative cloud infrastructures, devised based on established technologies of several major players on the cloud market, e.g., Amazon, Microsoft, Google, Cisco, VMware, and OpenStack. We also provide details on the hardware and software components used in the data center to manage the cloud services. We then apply several popular threat modeling methods on such cloud infrastructures, including attack surface, attack tree, attack graph, and security metrics based on attack trees and attack graphs.

The main contribution of this paper is twofold. First, to the best of our knowledge, this is the first comprehensive study of threat modeling based on well established models and concrete cloud data center designs, which incorporate technologies used by major cloud providers on the market. Second, our study provides answers to many practical questions, such as, *How can cloud providers gather and organize knowledge concerning the security of their cloud data center and services? How can cloud providers examine the security of a cloud data center at different abstraction levels? How can cloud providers measure the security of their cloud data center before and after applying a hardening option?* Those threat modeling efforts can not only provide cloud providers practical lessons and means for understanding and improving their cloud infrastructures, but may also imbed more confidence in cloud tenants by providing them a clearer picture about potential threats in cloud infrastructures.

The remainder of this paper is organized as follows. Section 2 provides the background knowledge on threat modeling and security metrics needed later in our work. In Section 3, the cloud data center architecture is presented. In Section 4, the threat modeling is explained in details. In Section 5, security metrics are applied to measure the level of security. Related work are reviewed in Section 6, and the paper concluded in Section 7.

2 Background

The following briefly reviews the threat models and security metrics that are applied in this paper, including attack surface, attack tree, attack graph, attack tree-based metric (ATM), and Bayesian network (BN)-based metric.

– Attack surface: Originally proposed as a metric for software security, attack surface captures software components that may lead to potential vulnerabilities. These may include entry and exit points (i.e., methods in a software program that either take user inputs or generate outputs), communication channels (e.g., TCP or UDP), and untrusted data items (e.g., configuration files or registry keys read by the software) [15]. Due to the complexity of examining source code, most existing work applies the concept in a less formal manner. For example, between an end user, the cloud provider, and cloud services, six attack surfaces can be composed [11].

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

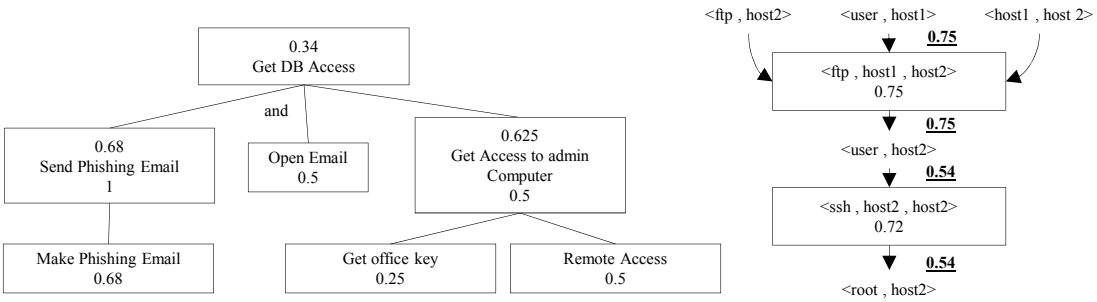


Fig. 1: Attack Tree (Left) and Attack Graph (Right)

– Attack tree: While attack surface focuses on what may provide attackers initial privileges or accesses to a system, attack trees demonstrate the possible attack paths which may be followed by the attacker to further infiltrate the system [20]. The left-hand side of Figure 1 shows an attack tree example in which the attacker's goal is to get accesses to the database. In the example, there are two ways to reach the root node (the goal). First, the attacker can follow the left and middle paths at the same time (due to the *and* label), or the attacker can follow the right path for reaching the root node.

– Attack graph: As a more fine-grained model, an attack graph depicts all possible attack steps and their causal relationships [22]. In the right-hand side of Figure 1, each triplet inside a rectangle indicates an exploit <service vulnerability, source host, destination host>, and each pair in plaintext indicates a pre- or post-condition <condition, host> of the exploits. The logic relationships between the nodes are represented based on the assumption that any exploit can be executed if and only if all of its pre-conditions are already satisfied (e.g., In Figure 1, the first exploit requires all three pre-conditions to be satisfied), whereas any condition may be satisfied by one exploit for which the former is a post-condition.

– The above threat models are all qualitative in nature. The attack tree-based metric (ATM) quantifies the threat in an attack tree using the concept of *probability of success* [8]. The probability of each node in the attack tree is typically determined based on historical data, expert opinions, or both. In Figure 1, a number above the label represents the overall probability of success, and a number below the label represents the probability of each node alone. The probability on the root node indicates the most risky path, which should be prioritized in security hardening. The BN-based metric [24, 9] can be applied to attack graphs to calculate the probability for an average attacker to compromise a critical asset. The conditional probabilities that an exploit can be executed given its pre-conditions are all satisfied can usually be estimated based on standard vulnerability scores (e.g., the CVSS scores [16]). In Figure 1, the probability inside a rectangle is the CVSS score divided by 10, and each underlined number represents the probability for successfully executing that exploit. In this example, the attack goal has a probability of 0.54, and if we change the ftp service on host2 and suppose the new probability becomes 0.4, then the new attack probability for the goal will become 0.228, indicating increased security.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3 Devising Cloud Data Center Infrastructures

In this section, we devise two cloud data center infrastructures that will be used for threat modeling in Section 4 and Section 5. To make our infrastructures more representative, we have base our infrastructures upon concepts and ideas borrowed from major players on the market, including Cisco, VMware, and OpenStack, as follows.

- Cisco presents a cloud data center design for both public and private clouds [4], which is divided into multiple layers with suggested hardware for the physical network and software used to virtualize the resources. We borrow the multi-layer concept and some hardware components, including Carrier Routing System (CRS), Nexus (7000,5000,2000), Catalyst 6500, and MDS 9000.
- VMware vSphere suggests the hardware and software components to run a private cloud data center [12]. They also tag the port numbers used to connect services together. We borrow the concepts of Authentication Server, Domain Name System(DNS), and Storage Area Network (SAN) and synthesize these to represent the main functionality of some hardware components in our cloud infrastructures.
- OpenStack is one of the most popular open source cloud operating systems [17]. We borrow following components of OpenStack: Dashboard, Nova, Neutron, Keystone, Cinder, Swift, Glance, and Ceilometer [17].

Table 1 compares the main concepts used in our infrastructures to the major cloud providers in the market, including Amazon [5], Microsoft [23], and Google [3] (some of those concepts will be discussed later in this section).

	AWS	Microsoft Azure	Google Compute
Multiple Layers	×	×	×
Authentication Sever	×	×	
Domain Name System	×	×	×
One service in each cluster	×	×	×
Multi-tier	×	×	×

Table 1: Concepts Used by Major Cloud Providers

We discuss two different infrastructures since OpenStack components can either run centrally on a single server or be distributed to multiple servers [17].

Infrastructure 1 Figure 2 illustrates our first infrastructure, which is based on concepts and technology presented by Cisco [4], VMware vSphere [12], and OpenStack [17]. The physical network provides accesses to both cloud users and cloud administrators. Cloud administrators connect to the data center through firewalls ($node$ 17) and ($node$ 19), an authentication server ($host$ 18), and Nexus 7000 ($node$ 20), which is connected to the other part of the network. For cloud users, Cisco's multi-layer concept is used [4] as follows.

- In Layer 1, a CRS ($node$ 1) is used to connect the cloud to the internet, which then connects to a firewall ($node$ 2, ASA 5500-X Series) while simultaneously being connected to two different types of servers (authentication servers ($host$ 3) as well as DNS and Neutron Servers ($node$ 4)). Those servers provide services to the cloud tenants and end users. The servers then connect to Cisco Nexus 7000 with Catalyst 6500 ($node$ 5) to route the requests to destination machines.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

– In Layer 2, a firewall ($node$ 6, ASA 5500-X Series) connects the first layer to this layer through Nexus 5000 ($node$ 7). The Nexus 5000 is used to connect rack servers through Nexus 2000, which is used to connect servers inside each rack at the computing level ($hosts$ $8, 9, 10, 11,$ and 12). The Nexus 5000 ($node$ 7) then connects to the next layer.
– In Layer 3, another Nexus 7000 ($node$ 13) connects the previous layer to the storage. A firewall ($node$ 14, ASA 5500-X Series) connects the Nexus 7000 ($node$ 13) and MDS 9000 ($node$ 16).

The following outlines how the cloud works. OpenStack components run on the authentication servers among which one ($host$ 3) is designated for cloud tenants, and another ($host$ 18) for cloud administrators. The first runs following components: Dashboard, Nova, Neutron, Keystone, Cinder, Swift, Glance, and MySql. The second runs the same components, but additionally runs Ceilometer for a billing system. The DNS server ($node$ 4) runs a Neutron component that provides the address of the machine running a requested service. At the computing level ($hosts$ $8, 9, 10, 11,$ and 12), all physical servers run four components: Hypervisor, Nova to host and manage VMs, Neutron agent to connect VMs to the network, and Ceilometer agent to calculate the usage. At the computing level, each physical server cluster runs the same VMs service, e.g., all $http$ VMs run on the $http$ server cluster, and the same occurs for application VMs, ftp VMs, smtp VMs, and database VMs. Finally, all physical machines and VMs run ssh for maintenance.

Infrastructure 2 The second infrastructure is illustrated in Figure 3, which is based on concepts and technology presented by Cisco [4], VMware vSphere [12], and OpenStack [17]. This infrastructure has a similar physical network as the previous, with the addition of new machines that separate OpenStack components, which are installed on the authentication servers for cloud tenants in the previous infrastructure, into many different machines. These new machines are Neutron servers ($node$ 25), controller servers ($node$ 36), and network nodes ($node$ 34). In addition, the authentication server ($host$ 23) for cloud tenants will run a Dashboard component to access and manage the VMs related to the tenant user. Moreover, Neutron server ($node$ 25) serves to control the virtual network and connects to the controller node ($node$ 36), which runs Nova API, Neutron API, Keystone, Glance, Swift, Cinder, MySql, and any component needed to manage and control the cloud. The last node is a network node ($node$ 34) which translate between the virtual IPs and the physical IPs to grant accesses to services running on VMs. For example, if a cloud tenant wishes to access their VMs, they will first need to connect to the Dashboard. Next, the Neutron server will send the authentication request to the keystone service on the Controller node. If the user possesses the privilege to access the VM, the controller will send a request to the network node to obtain the address for the VMs, and will then send the address to the Neutron server to connect the user to their VMs.

4 Threat Modeling

This section conducts threat modeling on the two infrastructures that are just introduced.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

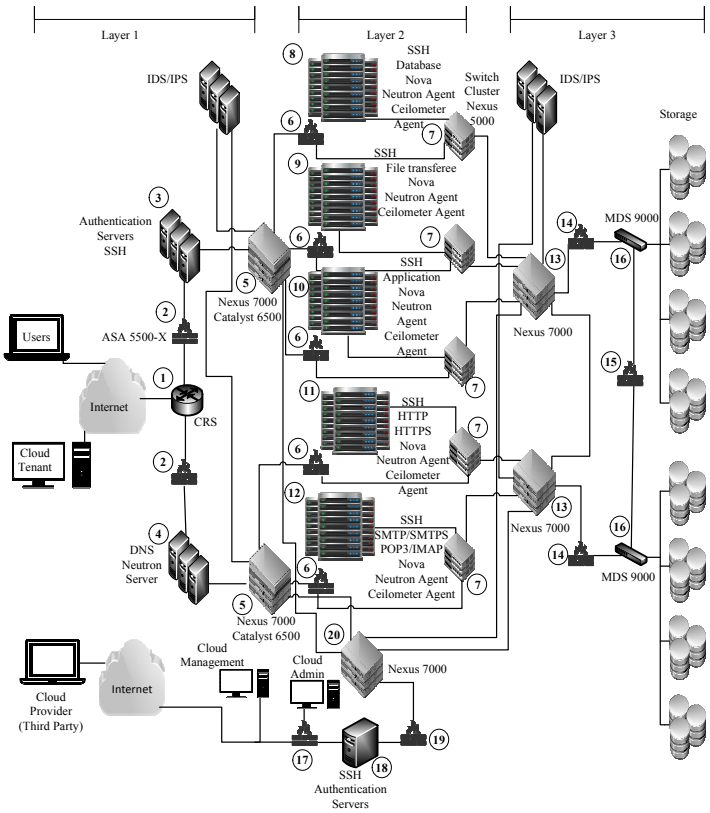


Fig. 2: Cloud Data Center Infrastructure 1

4.1 Attack Surface

In this section, we apply the attack surface concept at the resource level. Gruschka & Jensen categorize attack surfaces into those between user, service, and cloud provider [11]. The same classes are used in our discussions, with the addition of surfaces belonging to the same class. Also, we consider the service class used by Gruschka & Jensen [11] as the intermediate layer between users and the cloud provider in the sense that, if a user wishes to attack a cloud provider, he/she must pass through an attack surface consisting of services. In addition, we focus on entry and exit points [15] which indicate the means through which the attack starts and those through which data is leaked out, respectively.

In Figure 2 and 3 it can be observed that there are three types of attack surfaces in a cloud data center. First, there are attack surfaces related to the physical network, involving hardware and software components, such as switches, routers, servers, applications, and operating systems. Second, there are virtualization-related attack surfaces, such as hypervisors and virtual switches. Last, there are cloud operating systems, such as OpenStack components (Glance, Neutron, Nova, Ceilometer, and Keystone). The first type of attack surface is similar to those in traditional networks, but components related to cloud running on top of the physical network must also be considered. On the other

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

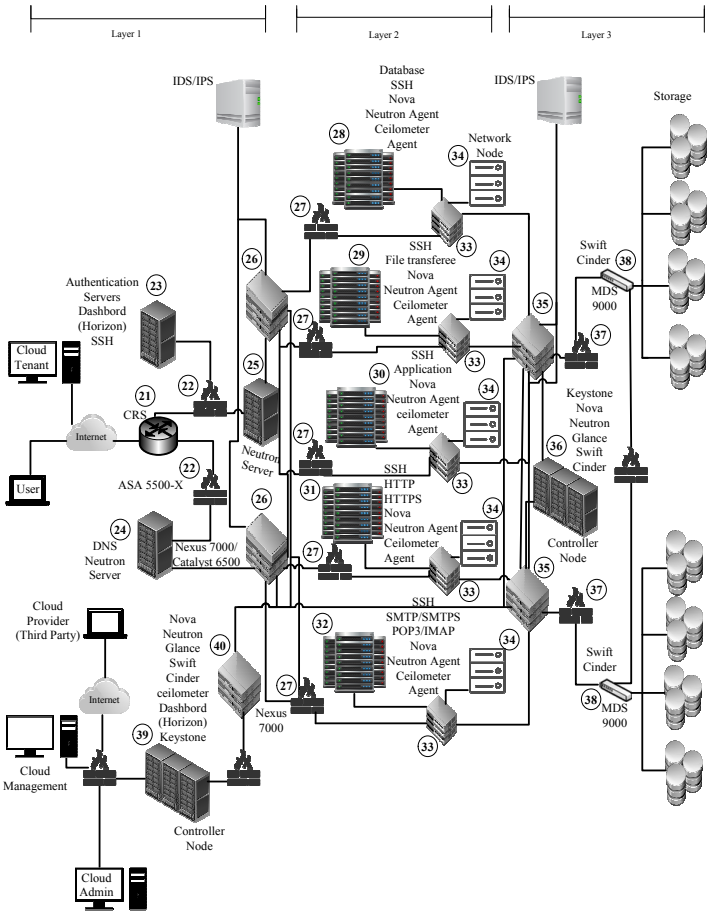


Fig. 3: Cloud Data Center Infrastructure 2

hand, virtualization and cloud operating systems-related attack surfaces are unique to cloud and their analysis will pose new challenges.

Attack Surface w.r.t. Users We consider two types of users. First, the normal user using the cloud service may aim to attack either the cloud tenant who owns that service, another cloud tenant or its users using the same cloud, or the cloud provider. Second, the cloud tenant may aim to attack another cloud tenant and its users, or the cloud provider. Various surfaces can be utilized by users to attack the cloud, including the hypervisor, VMs, APIs and web services, and OpenStack components (e.g., Horizon, Keystone, Neutron, Glance, and Nova).

Example 1. A normal user wants to attack a hypervisor on the database VM server ($host$ 8) to steal information about all VMs running on that machine. First, the entry point to start this attack is the database VM on the hypervisor. After he/she get initial accesses to the database VM, that VM become an exit point to attack the hypervisor. Finally, with accesses to the hypervisor, e.g., through exploiting CVE-2013-4344 [1], the

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

attacker can get data related to all VMs run on top of this hypervisor and the hypervisor thus becomes an exit point. Next consider a cloud tenant who wants to attack another tenant hosted on the same physical machine. First, the attacker can use his/her VM as entry point to get a privilege to the hypervisor, e.g., by applying CVE-2012-3515 [1], then the attacker will use the hypervisor as an entry point to get accesses to the target VM.

Attack Surface w.r.t. Cloud Providers A cloud provider here refers to an operator who has privileges to access certain components (e.g., switches, firewall, and SAN) for maintenance and management purposes. This type of attackers may use his/her accesses to resources to attack the cloud data center. All three types of attack surfaces explained before can be used by such an attacker.

Example 2. An operator who has accesses to Nexus 7000 ($node$ 13) for management wants to get accesses to sensitive data related to a tenant. First, he/she can use the Nexus 7000 as an entry point to obtain a root privilege on Nexus 7000, and then use this machine as an exit point to start another attack to get data from the storage device ($node$ 16).

4.2 Attack Tree

The previous section shows how attack surface may capture the initial attack attempts. To further study what may happen once an attacker gains initial privileges, we will need attack trees, which represent high level attack paths leading attackers to their goals. Figure 4 shows an attack tree for our cloud data center infrastructures. It is assumed that the root node, or goal node, is a storage device in the cloud that is susceptible to attacks by either a malicious user, a cloud tenant, or a cloud operator. Eight paths in Figure 4 represent the possible ways to reach such a target. Each path represents a capability level of users who can follow the path; not all paths can be used by all users. For example, some paths can be followed by the cloud operator but cannot be accessed by normal users or cloud tenants. In what follows, the paths and corresponding users will be explained in further details.

- **Path 1:** This attack can be executed by a normal user to obtain data from the storage device ($node$ 16). The user must first establish a connection to the $http$ VM server ($host$ 11) and must then acquire the root privilege on this VM. The attacker can then connect to the application VM server ($host$ 10) provided that they have obtained root privilege on that VM. After the user acquires access to the application VM, he/she may create a connection to the database VM server ($host$ 8). From this point, the user can attack the database VM to obtain root privilege on that VM. Finally, the attacker can launch an attack on the hypervisor to gain access to other database VMs ($host$ 8) running on the same physical machine and obtain data related to all database VMs stored on the storage device ($node$ 16).
- **Path 2:** The normal user can use this path to attack the cloud storage device ($node$ 38). The attacker begins the attack by surpassing the firewall ($node$ 22) to obtain privilege on OpenStack ($node$ 36) in order to gain a direct connection to the database VM server ($host$ 28). The remainder of this attack is similar to that of path 1, and serves to gain access to the hypervisor and the storage device.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

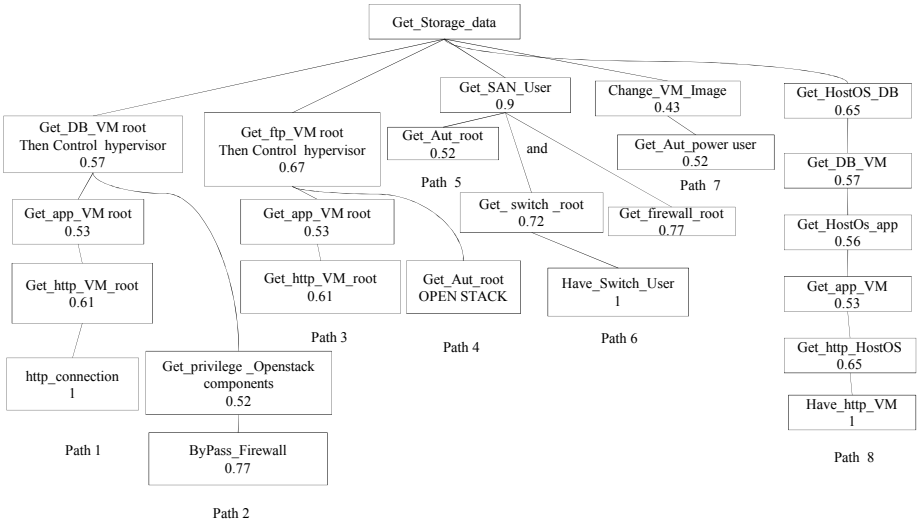


Fig. 4: Attack Tree

– **Path 3:** This path can be used by a cloud tenant user who has user access to the $http$ VM server ($host$ 11) and wishes to access ftp files stored on the storage device ($nose$ 16). First, the cloud tenant user must obtain root privilege on the $http$ VM server ($host$ 11). Then, he/she will need to obtain root privilege on the application VM server ($host$ 10) to start a connection to the ftp VM server ($host$ 9). After this, the user will obtain root privilege on this VM and get the ftp files related to this VM. In addition, the user can attack the hypervisor to obtain the ftp files related to other VMs running on top of this hypervisor.
– **Path 4:** Cloud tenants who do not already possess ftp VM servers running on the cloud can use this path to obtain data from the storage device ($node$ 16) through the ftp VM server ($host$ 9). Cloud tenants on this path will use OpenStack components ($host$ 3) to gain privileges to access the ftp VM ($host$ 9) belonging to another cloud tenant. In this situation, the attacker can obtain all files belonging to this VM. Furthermore, the attacker may attack the hypervisor to gain access to other ftp VMs running on the same physical machine.
– **Path 5:** Cloud operators with access to the admin user authentication server ($host$ 18) can use this path by obtaining root access to the authentication server. They can then use this device to obtain root access on the SAN device ($node$ 16) to control the data stored on the storage device.
– **Path 6:** This path can be used by a cloud operator who has access to a physical machine (e.g., a switch, firewall, or other type of machines) to attack the storage device. Suppose the attacker has user access to a switch device ($node$ 13) for maintaining this device. The attacker can then obtain root access on this device as well as root access to a firewall device ($node$ 14) between the switch device and the SAN ($node$ 16). These two accesses may allow him/her to create a connection to the SAN device and subsequently attack the SAN in order to access the stored data.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

– **Path 7:** This path may be used by a third party cloud provider who has access to the authentication server ($host$ 18) of an administrator. The user must obtain root access on the authentication server and must then gain privilege on the VM image storage ($host$ 18) and ($node$ 16). In this case, the user may use this privilege to modify or change the VM images stored on Glance. This new image will have a backdoor that can be used by the attacker to gain access to all VMs with this image.
– **Path 8:** This path can be used by either a cloud tenant or a normal user. The goal for these attackers is to control the data belonging to other cloud tenants on the cloud. The attacker must first have access to the $http$ VM sever ($host$ 31) and must gain access to the Host Operating System (HOS) ($host$ 31). By gaining access to the HOS, the attacker can obtain access to all VMs running on this machine. The attacker may then gain access to all application VMs ($host$ 30) connected to all $http$ VMs to which they have access. Subsequently, the attacker gains access to the application VMs which may run on different physical machines; the attacker may then acquire access over all HOS related to those VMs ($host$ 30). The attacker can then gain root access to the database VM server ($host$ 28) in order to obtain the data stored on the storage device. The attacker may also gain access to all HOS running database VMs ($host$ 28).

4.3 Attack Graph

In the previous section, the attack tree shows how an attacker may follow an attack path to reach the goal. However, this is done at a higher abstraction level without details about specific vulnerabilities. We now construct attack graphs to represent specific exploits of vulnerabilities that can be use to reach the goal. Although we can apply the standard attack graph concept designed for traditional networks, special consideration must be given to the virtualization level, which is unique to cloud, and the fact that machines or VMs may have similar or identical configurations.

We construct our attack scenarios based on real vulnerabilities related to hardware and software components used in our infrastructures as listed in the National Vulnerability Database (NVD) [1]. In our attack graphs, the Common Vulnerability Scoring System (CVSS) [16] scores retrieved from the NVD are depicted inside each node after dividing it by 10 to obtain a probability value between 0 to 1, which is later used in the BN-based metric. An attack graph may be created for different types of users but we will focus on the normal user due to space limitations.

Figure 5 and Figure 6 show two attack graphs for the data center infrastructures depicted in Figure 2 and Figure 3, respectively. It is assumed that the attacker has access to a cloud tenant's services. The main goal for the attacker is to steal data from the storage. The user must have access to the $http$ VM as well as the application VM and database VM before reaching the goal due to the multi-tier infrastructure. The following services are assumed to be used in the data centers.

– Tectia Server version 5.2.3, for ssh running in all VMs.
– Apache $http$ server running on $http$ VM.
– Oracle version 10.1.0.2 installed on the application VM.
– Oracle version 10.2.1 on the database VM.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

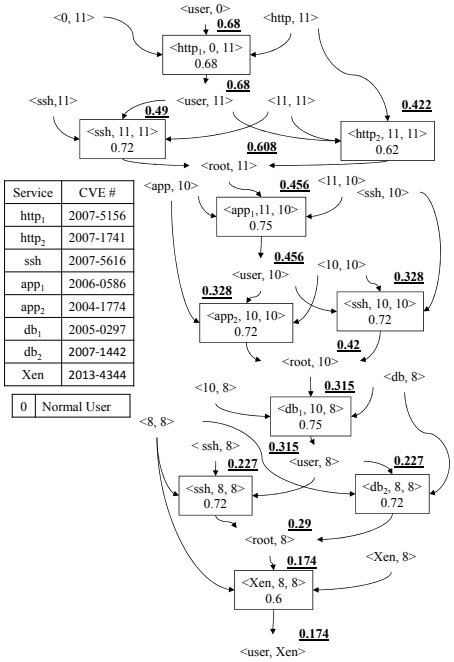


Fig. 5: Attack Graph for Figure 2

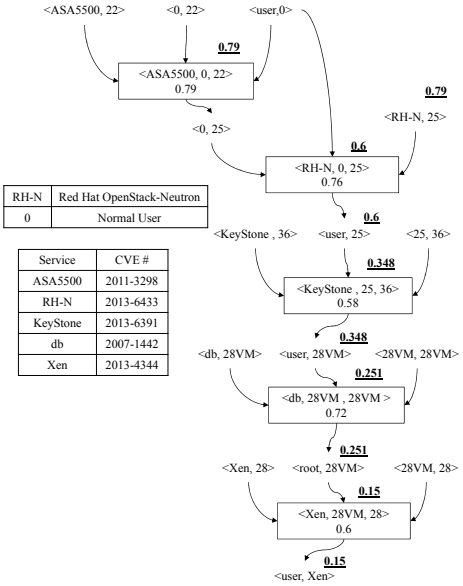


Fig. 6: Attack Graph for Figure 3

– Xen version 4.3.0 is running as a hypervisor to control VMs on physical machines.

Example 3. Figure 5 shows an attack graph corresponding to path 1 in the aforementioned attack tree. Between five to seven vulnerabilities are required to reach the goal. Specifically, five vulnerabilities are required if we assume the ssh vulnerability will be the same in the $http$ server VM, application server VM, and database server VM, whereas seven vulnerabilities are required if the ssh vulnerability is not used to reach the goal. We divide the attack graph to four stages and in each stage the attacker will gain a different level of privileges.

– **Stage 1:** A vulnerability in the $http$ server VM ($host$ 11) (CVE-2007-5156) is employed by the attacker to gain user access by uploading and executing arbitrary code containing .php. in the file extension as well as unknown extensions. Then, another vulnerability on the same VM (CVE-2007-1741) is used to gain root privilege by renaming the directory or performing symlink. A ssh ($host$ 11) vulnerability (CVE-2007-5156) can also be used to gain root privilege on the same VM.
– **Stage 2:** The attacker now can connect to the application server ($host$ 10). Then, by using a vulnerability related to the application sever VM (CVE-2006-0586), the attacker is allowed to gain the user privilege by executing arbitrary sql commands through multiple parameters. To gain root privilege on this VM, the attacker can apply this vulnerability (CVE-2004-1774) or by using an ssh ($host$ 10) vulnerability (CVE-2007-5616), and at this point the attacker can start a connection to the database server VM.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

– **Stage 3:** The attacker uses a vulnerability related to the database server ($host$ 8) VM (CVE-2005-0297) to gain user access. Then, on this VM he/she can gain root access by using vulnerability (CVE-2007-1442) or an ssh ($host$ 8) vulnerability (CVE-2007-5616).
– **Stage 4:** The attacker can then obtain data related to this database VM ($host$ 8), and he/she may obtain even more data from another VM running on the same physical machine by gaining access to a hypervisor through exploiting (CVE-2013-4344).

Example 4. The attack graph in Figure 6 is related to the infrastructure shown in Figure 3, where OpenStack components run on more than one physical machine. The goal of this attack is to gain access to date storage in three stages. This attack graph corresponds to path 2 in the attack tree.

– **Stage 1:** A vulnerability in the firewall ($node$ 22) (CVE-2011-3298) is employed by the attacker to bypass the firewall in order to connect to the Neutron server ($node$ 25). The attacker can then use the Neutron vulnerability (CVE-2013-6433) to gain privileges with which he/she can use vulnerability (CVE-2013-6391) to generate EC2 token API in order to gain access to a database VM ($host$ 28).
– **Stage 2:** After the attacker obtains access to the database VM ($host$ 28), he/she used the database vulnerability (CVE-2007-1442) to gain root privilege on the same VM. This allows the attacker to obtain data related to this VM.
– **Stage 3:** To obtain data from another database on the same physical machine, the attacker used the vulnerability (CVE-2013-4344) to gain access to the hypervisor running on this physical machine such that he/she can access all VMs running on this machine and view the data related to these VMs.

By constructing the attack surface, attack tree, and attack graphs for the cloud data center infrastructures, we have demonstrated how each model may capture potential threats at a different abstraction layer. Nonetheless, all those models are qualitative in nature, and we will apply security metrics to measure the threats in the coming section.

5 Cloud Security Metric

In this section, we apply two security metrics based on the attack tree and attack graphs, respectively, to further quantify the threats modeled in the previous section.

5.1 Attack Tree Metric

In this section, an attack tree metric (ATM) will be applied based on the attack tree described in Section 4.2. In Figure 7, all nodes inside the same path are considered as having AND relationships, whereas an OR relationship is assumed between different paths unless if an AND relationship is explicitly stated. Based on such assumptions, the corresponding equations are applied to calculate the probabilities. The highest probability is assigned to the root node after applying the metric. In Figure 7, between the two probabilities in each node, the probability with (+) represents the average CVSS values and the other probability represents the metric result.

In Figure 7, it can be observed that path 5 and 6 are the least secure paths in the attack tree. Those two paths can be followed by a cloud operator to launch an insider

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

attack to steal data from the storage device. This metric can also be used to verify whether or not adding a new service or disabling existing services can increase security and by how much. As shown in Figure 7, the probability to reach n_8 is 0.45; as such, if the cloud provider wishes to decide whether to increase security levels in that node, he/she can use the metric before and after applying the changes. For example, suppose the cloud provider wishes to add new rules to a firewall to prevent attacks from n_9 and n_{11} to n_8. After re-applying the ATM metric, the probability on n_8 becomes 0.348, showing increased security. Applying the ATM on other potential changes may help the cloud provider to make the right decisions in hardening the cloud.

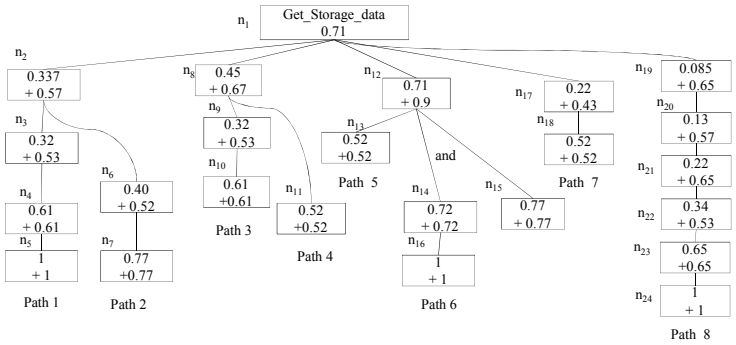


Fig. 7: Attack Tree Metric

5.2 Bayesian Network Metric

In this section, the BN-based security metric [24, 9] will be applied to the attack graph shown in Figure 5 to measure the threat and also the effect of certain changes made to the infrastructure. In particular, we show how the level of redundancy and diversity may affect the security of the cloud infrastructure. For redundancy, the ssh service running on some of the servers will be disabled to see the effect on security. As to diversity, we assume the ssh service may be diversified with other software, e.g., OpenSSH version 4.3, denoted as ssh_2, with a vulnerability CVE-2009-290 and a CVSS score of 6.9 [1].

Table 2 shows how security is affected by reducing redundancy and increasing diversity through disabling or diversifying some of the ssh instances in the infrastructure. In the left-hand side table, the first row shows that the probability for an attacker to reach the goal is 0.174 in the original configuration, and the remaining rows show the same probability after disabling one or more ssh instances on the three servers, e.g., the probability after disabling ssh on the $http$ server is reduced to 0.121, which corresponds to the most secure option by disabling one ssh instance, and the lowest probability after disabling two and three ssh instances is 0.094 and 0.074, respectively.

The middle and right-hand side of Table 2 show the effect of diversifying the ssh instances. In the middle figure, we can observe that, after we replace the ssh service on app and DB servers with ssh_2, the probability for reaching the goal decreases from 0.174 to 0.171, which indicates a slight improvement in security. The next three rows of

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

the table show that the same effect remains when one of the ssh instances is disabled. The last three rows show the simple fact that, when there is only one ssh instance left, the diversification effort has not effect.

In the right-hand side of Table 2, we change the ssh instance on the $http$ server instead of the app server, as in the above case, in order to see whether different diversification options make any difference to security. We can see the probability decreases in most cases (except the fourth row), which indicates a slightly more effective option than the previous one. Overall, the best option in terms of diversification without disabling any service instance is given in the first row in the right table, with a probability 0.17, and the best option for disabling one service instance is given in the fourth row of the middle table with a probability 0.119 (disabling two instances always yields 0.094). Obviously, more options may be evaluated similarly using the BN-based metric in order to find the best option for making the cloud data center infrastructure more secure.

$\langle user, Xen \rangle$			
$http$	app	DB	T
ssh			T
T	T	T	0.174
T	F	T	0.136
T	T	F	0.136
F	T	T	0.121
T	F	F	0.106
F	F	T	0.094
F	T	F	0.094
F	F	F	0.074

$\langle user, Xen \rangle$			
$http$	app	DB	T
ssh_1	ssh_2	ssh_2	T
T	T	T	0.171
T	F	T	0.135
T	T	F	0.135
F	T	T	0.119
T	F	F	0.106
F	F	T	0.094
F	T	F	0.094
F	F	F	0.074

$\langle user, Xen \rangle$			
$http$	app	DB	T
ssh_2	ssh_1	ssh_2	T
T	T	T	0.17
T	F	T	0.133
T	T	F	0.134
F	T	T	0.12
T	F	F	0.105
F	F	T	0.094
F	T	F	0.094
F	F	F	0.074

Table 2: The BN-Based Metric Results for the Attack Graph Shown in Figure 5

6 Related Work

Cloud environments are usually subject to many security threats some of which exploit existing vulnerabilities related to the cloud [10]. There only exist limited efforts on threat modeling for cloud infrastructures. Ingalsbe et al. present a threat model that cloud tenants can use to evaluate the system [13]. The authors adopt an Enterprise Threat Modeling methodology, which classify all components related to the cloud tenant under three categories (Actor, End Points, and Infrastructure). However, the authors do not provide concrete case studies detailing how such a threat model might be used. Gruschka & Jensen apply the attack surface concept to provide classifications for attacks in a cloud [11]. The authors identify three main entities (User, Cloud provider, and Service) and the attack surfaces between those entities. The authors provide high level examples of attacks but do not mention specific services or vulnerabilities underlying each attack surface. We borrow this classification in devising our threat models. The original attack surface concept [15] is intended to measure the security of a software system focusing on identifying entry/exit points, communication channels, and untrusted data items from the source code. Like most existing work, our work applies those concepts but at a higher abstraction level.

Attack tree is a well known threat model which can be used for many useful analyses, such as analyzing the relative cost of attacks and the impact of one or more attack vectors [20]. Attack trees can also be used in security hardening to determine the best

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

options to increase security within a budget [7]. Using attack trees can help to understand what kind of attackers may follow an attack tree path [20, 18]. Attack graphs can be automatically generated by modeling the network and vulnerabilities, and many useful analyses may be performed using attack graphs [22]. We borrow the concepts of attack trees and attack graphs but apply them to cloud data center infrastructures that we have devised. There exist many research work on extending attack trees and attack graphs to security metrics. A probabilistic metric is applied to attack graphs to obtain an overall attack likelihood for the network [24]. Edge et al. presented protection trees [8] which are similar to attack trees but contain information on how the system can be secured, and our work borrows part of this work to apply the attack tree-based metric. A BN-based security metric applies attack graphs to measure the security level of a network [9]. The metric converts the CVSS scores of vulnerabilities into attack probabilities and then obtain the overall attack likelihood for reaching critical assets. We apply this metric to our cloud data center infrastructures in this paper. The National Institute of Standards and Technology (NIST) underline the importance of security measuring and metrics for cloud providers by providing high level definitions and requirements but no concrete methodologies [2]. Luna et al. propose a framework with basic building blocks for cloud security metrics [14]. We loosely follow the framework in this paper.

7 Conclusion

In this paper, we have conducted threat modeling and measuring for cloud data center infrastructures. First, we have shown two cloud data center infrastructures which are fictitious but represent many existing technologies adopted at real cloud data centers by major cloud providers. Three threat models were then applied to those infrastructures, namely, the attack surface, attack trees, and attack graphs, which model potential threats from different viewpoints and at different abstraction levels. We have also applied security metrics based on attack trees and attack graphs, respectively, to quantify the threats. This work will benefit cloud providers in demonstrating how threat models and metrics may assist them in evaluating and improving the security of their clouds. Future work will focus on extending the scale and scope of our existing efforts and developing automated hardening algorithms for cloud data centers to generate actionable knowledge from the threat modeling and measuring results.

Disclaimer This paper is not subject to copyright in the United States. Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

1. National vulnerability database. `http://www.nvd.org`. [Online; accessed 20/02/2015].
2. National Institute of Standards and Technology: Cloud Computing Service Metrics Description. `http://www.nist.gov/itl/cloud/upload/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf`, 2015. [Online; accessed 17/06/2015].

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3. B. Adler. Google Compute Engine Performance Test with RightScale and Apica. http://www.rightscale.com/blog/cloud-industry-insights/google-compute-engine-performance-test-rightscale-and-apica, 2013. [Online; accessed 26/03/2016].
4. K. Bakshi. Cisco cloud computing-data center strategy, architecture, and solutions. *DOI= http://www. cisco. com/web/strategy/docs/gov/CiscoCloudComputing_WP. pdf*, 2009.
5. J. Barr. Building three-tier architectures with security groups. https://aws.amazon.com/blogs/aws/building-three-tier-architectures-with-security-groups/, 2010. [Online; accessed 28/03/2016].
6. K. Dahbur, B. Mohammad, and A. B. Tarakji. A survey of risks, threats and vulnerabilities in cloud computing. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications*, ISWSA '11, pages 12:1–12:6, New York, NY, USA, 2011. ACM.
7. R. Dewri, I. Ray, N. Poolsappasit, and D. Whitley. Optimal security hardening on attack tree models of networks: a cost-benefit analysis. *International Journal of Information Security*, 11(3):167–188, 2012.
8. K. S. Edge, G. C. Dalton, R. A. Raines, and R. F. Mills. Using attack and protection trees to analyze threats and defenses to homeland security. In *MILCOM 2006 - 2006 IEEE Military Communications conference*, pages 1–7, Oct 2006.
9. M. Frigault and L. Wang. Measuring network security using bayesian network-based attack graphs. In *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, pages 698–703, July 2008.
10. B. Grobauer, T. Walloschek, and E. Stöcker. Understanding cloud computing vulnerabilities. *Security & privacy, IEEE*, 9(2):50–57, 2011.
11. N. Gruschka and M. Jensen. Attack surfaces: A taxonomy for attacks on cloud services. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 276–279, July 2010.
12. M. Hany. VMware VSphere In The Enterprise. http://www.hypervizor.com/diags/HyperViZor-Diags-VMW-vS4-Enterprise-v1-0.pdf. [Online; accessed 05/02/2015].
13. J. A. Ingalsbe, D. Shoemaker, and N. R. Mead. Threat modeling the cloud computing, mobile device toting, consumerized enterprise-an overview of considerations. In *AMCIS*, 2011.
14. J. Luna, H. Ghani, D. Germanus, and N. Suri. A security metrics framework for the cloud. In *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*, pages 245–250, July 2011.
15. P. Manadhata and J. Wing. An attack surface metric. *Software Engineering, IEEE Transactions on*, 37(3):371–386, May 2011.
16. P. Mell, K. Scarfone, and S. Romanosky. Common vulnerability scoring system. *IEEE Security & Privacy*, 4(6):85–89, 2006.
17. Openstack. Openstack Operations Guide. http://docs.openstack.org/openstack-ops/content/openstack-ops_preface.html. [Online; accessed 27/08/2015].
18. I. Ray and N. Poolsapassit. *Computer Security – ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005. Proceedings*, chapter Using Attack Trees to Identify Malicious Attacks from Authorized Insiders, pages 231–246. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
19. P. Saripalli and B. Walters. Quirc: A quantitative impact and risk assessment framework for cloud security. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 280–288, July 2010.
20. B. Schneier. Attack trees. *Dr. Dobb's journal*, 24(12):21–29, 1999.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

21. F. B. Shaikh and S. Haider. Security threats in cloud computing. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 214–219, Dec 2011.
22. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 273–284, 2002.
23. R. Squillace. Azure infrastructure services implementation guidelines. `https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-infrastructure-service-guidelines/`, 2015. [Online; accessed 28/03/2016].
24. L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In V. Atluri, editor, *Data and Applications Security XXII*, volume 5094 of *Lecture Notes in Computer Science*, pages 283–296. Springer Berlin Heidelberg, 2008.

Alhebaishi, Nawaf; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Threat Modeling for Cloud Data Center Infrastructures."
Paper presented at 9th International Conference on Foundations and Practice of Security,
Quebec City, Canada. October 24, 2016 - October 26, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Simpira v2: A Family of Efficient Permutations Using the AES Round Function

Shay Gueron[1,2] and Nicky Mouha[3,4,5]

[1] Department of Mathematics, University of Haifa, Israel.
[2] Intel Corporation, Israel Development Center, Haifa, Israel.
[3] Dept. Electrical Engineering-ESAT/COSIC, KU Leuven, Leuven and iMinds, Ghent, Belgium.
[4] Project-team SECRET, Inria, France.
[5] National Institute of Standards and Technology, Gaithersburg, MD, USA.
shay@math.haifa.ac.il, nicky@mouha.be

Abstract. This paper introduces Simpira, a family of cryptographic permutations that supports inputs of $128 \times b$ bits, where b is a positive integer. Its design goal is to achieve high throughput on virtually all modern 64-bit processors, that nowadays already have native instructions for AES. To achieve this goal, Simpira uses only one building block: the AES round function. For $b = 1$, Simpira corresponds to 12-round AES with fixed round keys, whereas for $b \geq 2$, Simpira is a Generalized Feistel Structure (GFS) with an F-function that consists of two rounds of AES. We claim that there are no structural distinguishers for Simpira with a complexity below 2^{128}, and analyze its security against a variety of attacks in this setting. The throughput of Simpira is close to the theoretical optimum, namely, the number of AES rounds in the construction. For example, on the Intel Skylake processor, Simpira has throughput below 1 cycle per byte for $b \leq 4$ and $b = 6$. For larger permutations, where moving data in memory has a more pronounced effect, Simpira with $b = 32$ (512 byte inputs) evaluates 732 AES rounds, and performs at 824 cycles (1.61 cycles per byte), which is less than 13 % off the theoretical optimum. If the data is stored in interleaved buffers, this overhead is reduced to less than 1 %. The Simpira family offers an efficient solution when processing wide blocks, larger than 128 bits, is desired.

Keywords. Cryptographic permutation, AES-NI, Generalized Feistel Structure (GFS), Beyond Birthday-Bound (BBB) security, hash function, Lamport signature, wide-block encryption, Even-Mansour.

1 Introduction

The introduction of AES instructions by Intel (subsequently by AMD, and recently ARM) has changed the playing field for symmetric-key cryptography on modern processors, which lead to a significant reduction of the encryption overheads. The performance of these instructions has been steadily improving in every new generation of processors. By now, on the latest Intel Architecture

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Codename Skylake, the AESENC instruction that computes one round of AES has latency of 4 cycles and throughput of 1 cycle. The improved AES performance trend can be expected to continue, with the increasing demand for fast encryption of more and more data.

To understand the impact of the AES instructions in practice, consider for example the way that Google Chrome browser connects to `https://google.com`. In this situation, Google is in a privileged position, as it controls both the client and the server side. To speed up connections, Chrome (the client) is configured to identify the processor's capabilities. If AES-NI are available, it would offer (to the server) to use AES-128-GCM for performing authenticated encryption during the TLS handshake. The high-end server would accept the proposed cipher suite, due to the high performance of AES-GCM on its side. This would capture any recent 64-bit PC, tablet, desktop, or even smartphone. On older processors, or architectures without AES instructions, Chrome resorts to proposing the ChaCha20-Poly1305 algorithm during the secure handshake negotiation.

An advantage of AES-GCM is that the message blocks can be processed independently for encryption. This allows pipelining of the AES round instructions, so that the observed performance is dominated by their throughput, and not by their latency [43, 44]. We note that even if a browser negotiates to use an inherently sequential mode such as CBC encryption, the web server can process multiple independent data buffers in parallel to achieve high throughput (see [43, 44]), and this technique is already used in the recent OpenSSL version 1.0.2. This performance gain by collecting multiple independent encryption tasks and pipelining their execution, is important for the design rationale of Simpira.

Setting. This paper should be understood in the following setting. We focus only on processors with AES instructions. Assuming that several independent data sources are available, we explore several symmetric-key cryptographic constructions with the goal of achieving a high throughput. Our reported benchmarks are performed on the latest Intel processor, namely Architecture Codename Skylake, but we expect to achieve similar performance on any processor that has AES instructions with throughput 1.

In particular, we focus here on applications where the 128-bit block size of AES is not sufficient, and support for a wider range of block sizes is desired. This includes various use cases such as permutation-based hashing and wide-block encryption, or just to easily achieve security beyond 2^{64} input blocks without resorting to (often inefficient) modes of operation with "beyond birthday-bound" security. For several concrete suggestions of applications, we refer to Sect. 7.

Admittedly, our decision to focus on only throughput may result in unoptimized performance in certain scenarios where the latency is critical. However, we point out that this is not only a property of Simpira, but also of AES itself, when it is implemented on common architectures with AES instructions. To achieve optimal performance on such architectures, AES needs to be used in a parallelizable mode of operation, or in a protocol that supports processing

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

independent inputs. Similarly, this is the case for Simpira as well. In fact, for 128-bit inputs, Simpira is the same as 12-round AES with fixed round keys.

Origin of the name. Simpira is named after a mythical animal of the Peruvian Amazon. According to the legend, one of its front legs has the form of a spiral that can be extended to cover the entire surface of the earth [26]. In a similar spirit, the Simpira family of permutations extends itself to a very wide range of input sizes. Alternatively, Simpira can be seen as an acronym for "SIMple Permutations based on the Instruction for a Round of AES."

Update. This paper proposes Simpira v2. Compared to Simpira v1, the Type-1.x GFS by Yanagihara and Iwata was found to have a problem (see Sect. 8), and is replaced by a new construction that performs the same number of AESENCs. We also updated the round constants (see Sect. 4). Although no attack is currently known on Simpira v2 with the old rotation constants, the new constants seem to strengthen Simpira without affecting its performance in our benchmarks. Unless otherwise specified, Simpira in this document is assumed to refer to Simpira v2.

2 Related Work

Block ciphers that support wide input blocks have been around for a long time. Some of the earliest designs are Bear and Lion [2], and Beast [61]. They are higher-level constructions, in the sense that they use hash functions and stream ciphers as underlying components.

Perhaps the first wide-block block cipher that is not a higher-level construction is the Hasty Pudding Cipher [74], which supports block sizes of any positive number of bits. Another early design is the Mercy block cipher that operates on 4096-bit blocks [27]. More recently, low-level constructions that can be scaled up to large input sizes are the SPONGENT [17,18] permutations and the LowMC [1] block ciphers.

Our decision to use only the AES round function as a building block for Simpira means that some alternative constructions are not considered in this paper. Of particular interest are the EGFNs [7] used in Lilliput [6], the AESQ permutation of PAEQ [13], and Haraka[6] [55]. The security claims and benchmark targets of these designs are very different from those of Simpira. We only claim security up to 2^{128} blocks of input. However unlike Haraka, we consider all distinguishing attacks up to this bound. Also, we focus only on throughput, and not on latency. An interesting topic for future work is to design variants of these constructions with similar security claims, and to compare their security and implementation properties with Simpira.

[6] The first version of Haraka was vulnerable to an attack by Jérémy Jean [51] due to a bad choice of round constants. We therefore refer to the second version of Haraka, which prevents the attack.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3 Design Rationale of Simpira

AES [31] is a block cipher that operates on 128-bit blocks. It iterates the AES round function 10, 12 or 14 times, using round keys that are derived from a key of 128, 192 or 256 bits, respectively. On Intel (and AMD) processors, the AES round function is implemented by the `AESENC` instruction. It takes a 128-bit state and a 128-bit round key as inputs, and returns a 128-bit output that is the result of applying the `SubBytes`, `ShiftRows`, `MixColumns` and `AddRoundKey` operations. An algorithmic description of `AESENC` is given in Alg. 1 of Sect. 4, where we give the full specification of Simpira.

A cryptographic permutation can be obtained by setting the AES round keys to fixed, publicly-known values. It is a bad idea to set all round keys to zero. Such a permutation can easily be distinguished from random: if all input bytes are equal to each other, the AES rounds preserve this property. Such problems are avoided when round constants are introduced: this breaks the symmetry inside every round, as well as the symmetry between rounds. Several ciphers are vulnerable to attacks resulting from this property, such as the CAESAR candidate PAES [52,53] and the first version of Haraka [51]. The aforementioned design criterion, already present in Simpira v1, excludes the round constants of these designs.

We decided to use two rounds of AES in Simpira as the basic building block. As the `AESENC` instruction includes an XOR with a round key, this can be used to introduce a round constant in one AES round, and to do a "free XOR" in the other AES round. An added advantage is that two rounds of AES achieve *full bit diffusion*: every output bit depends on every input bit, and every input bit depends on every output bit.

Another design choice that we made, is to use only AES round functions in our construction, and no other operations. Our hope is that this design would maximize the contribution of every instruction to the security of the cryptographic permutation. It also simplifies the analysis and the implementation. From the performance viewpoint, the theoretically optimal software implementation would be able to dispatch a new `AESENC` instruction in every CPU clock cycle. A straightforward way to realize this design strategy is to use a (Generalized) Feistel Structure (GFS) for $b \geq 2$ that operates on b input subblocks of 128 bits each, as shown in Fig. 1.

As with any design, our goal is to obtain a good trade-off between security and efficiency. In order to explore a large design space, we use simple metrics to quickly estimate whether a given design reaches a sufficient level of security, and to determine its efficiency. In subsequent sections, we will formally introduce the designs, and study them in detail to verify the accuracy of our estimates.

3.1 Design Criteria

Our design criteria are as follows. The significance of both criteria against cryptanalysis attacks will be explained in Sect. 6.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

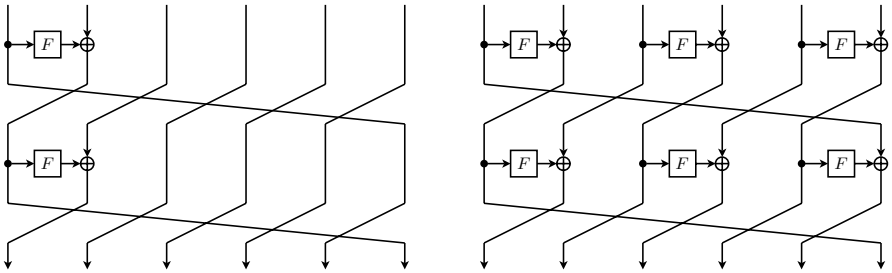


Fig. 1. Two common classes of Generalized Feistel Structures (GFSs) are the Type-1 GFS (left) and the Type-2 GFS (right). For each example, two rounds are shown of a GFS that operates on $b = 6$ subblocks. We will initially consider these GFSs in this paper, as well as other GFSs with a different number of F-functions per round, and other *subblock shuffles* at the end of every round. At a later stage, we will consider more advanced constructions as well.

- **Security**: We calculate the number of Feistel rounds to achieve either *full bit diffusion*, as well as the number of Feistel rounds to achieve at least 25 (linearly or differentially) active S-boxes. To ensure a sufficient security margin against known attacks, we require that the number of rounds is three times the largest of these two numbers.
- **Efficiency**: As explained in Sect. 1, we will only focus on throughput. Given that we use no other operations besides the AES round function, we will use the number of AES round functions as an estimate for the total number of cycles.

Suzaki and Minematsu [75] formally defined DRmax to calculate how many Feistel rounds are needed for an input subblock to affect all the output subblocks. We will say that *full subblock diffusion* is achieved after DRmax rounds of the permutation or its inverse, whichever is greater. To achieve the strictly stronger criterion of *full bit diffusion*, one or two additional Feistel rounds may be required.

To obtain a lower bound for the minimum number of active S-boxes, we use a simplified representation that assigns one bit to every pair of bytes, to indicate whether or not they contain a non-zero difference (or linear mask). This allows us to use the Mixed-Integer Linear Programming (MILP) technique introduced by Mouha et al. [69] to quickly find a lower bound for the minimum number of active S-boxes.

3.2 Design Space Exploration

For each input size of the permutation, we explore a range of designs, and choose the one that maximizes the design criteria. If the search returns several alternatives, it does not really matter which one we choose. In that case, we arbitrarily choose the "simplest" design. The resulting Simpira design is shown in Fig. 2–3.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

We will restrict ourselves to "simple" designs, such as for example constructions with identical round functions, instead of exhaustively searching for the optimal design that satisfies the design criteria. This is meant to simplify the cryptanalysis, as well as the implementation. We revisit this assumption in App. C.

Case $b = 1$. Full bit diffusion is reached after two rounds of AES, and four rounds of AES ensures at least 25 active S-boxes [31]. Following the design criteria, we select a design with 12 AES rounds.

Case $b = 2$. This is a (standard) Feistel structure. Full subblock diffusion is achieved after two Feistel rounds, and three Feistel rounds are needed to reach full bit diffusion. We find that five rounds ensures that there are at least 25 active S-boxes (see Fig. 5). Consequently, we select a design with 15 Feistel rounds.

Case $b = 3$. There are several designs that are optimal according to our criteria. They have either one or two F-functions per Feistel round, and various possibilities exist to reorder the subblocks at the end of every Feistel round. We choose what is arguably the simplest design: a Type-1 GFS according to Zheng et al.'s classification [82]. Full subblock diffusion requires five Feistel rounds, and at least six Feistel rounds are needed to ensure that there are least 25 active S-boxes. As seven Feistel rounds are needed to achieve full bit diffusion, we select a design with 21 Feistel rounds.

Case $b \geq 4$. The Type-1 GFS does not scale well for larger b, as diffusion becomes the limiting factor. More formally, Yanagihara and Iwata [78, 79] proved that the number of rounds required to reach full subblock diffusion is (at best) quadratic in the number of subblocks, regardless of how the subblocks are reordered at the end of every Feistel round.

In Simpira v1, the Yanagihara and Iwata's Type-1.x (b,2) GFS [80] was used for $b \geq 4$, except for $b = 6$ and $b = 8$. This is a design with two F-functions per round, where the number of rounds for full subblock diffusion is linear in b. Unfortunately, as we will explain in Sect. 8, this GFS is problematic as the same input subblock can be processed by more than one F-function. This general observation enabled attacks on Simpira v1 by Dobraunig et al. [35] and by Rønjom [73].

The Simpira v2 in this paper addresses this problem by ensuring that every subblock will enter an F-function only once. We do this by means of a new GFS construction. It uses $4b - 6$ F-functions to reach full bit diffusion, and ensures that at least 30 S-boxes are active (see also App. A). This construction is iterated three times, resulting in a design with $12b - 18$ F-functions, which is the same number of F-function as in Simpira v1.

We could also have used this construction for $b = 4$. However, we instead chose to go for a Type-2 GFS with 15 rounds. This not only results in a simpler

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

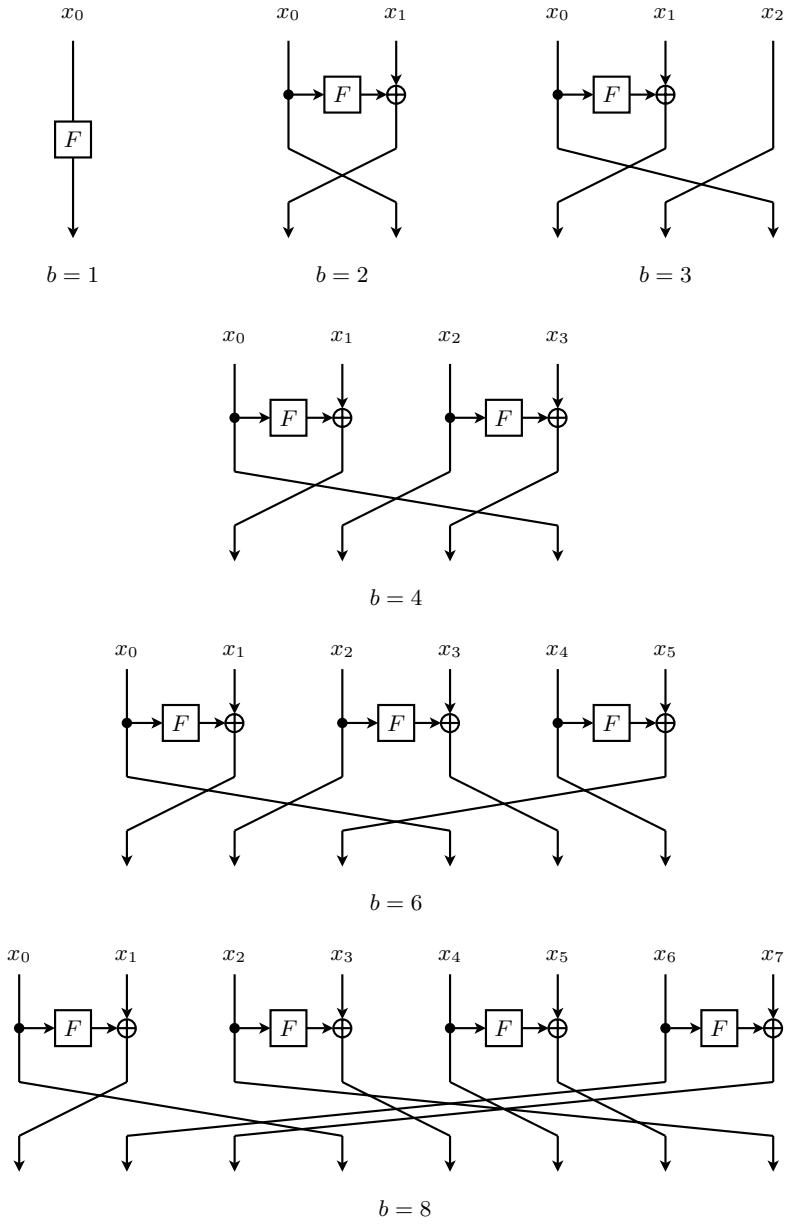


Fig. 2. One round of the Simpira construction for $b \in \{1, 2, 3, 4, 6, 8\}$. The total number of rounds is 6 for $b = 1$, 15 for $b = 2$, $b = 4$ and $b = 6$, 21 for $b = 3$, and 18 for $b = 8$. F is shorthand for $F_{c,b}$, where c is a counter that is initialized by one, and incremented after every evaluation of $F_{c,b}$. Every $F_{c,b}$ consists of two AES round evaluations, where the round constants that are derived from (c, b). The last round is special: the `MixColumns` is omitted when $b = 1$, and the final subblocks may be output in a different order. See Sect. 4 for a full specification.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

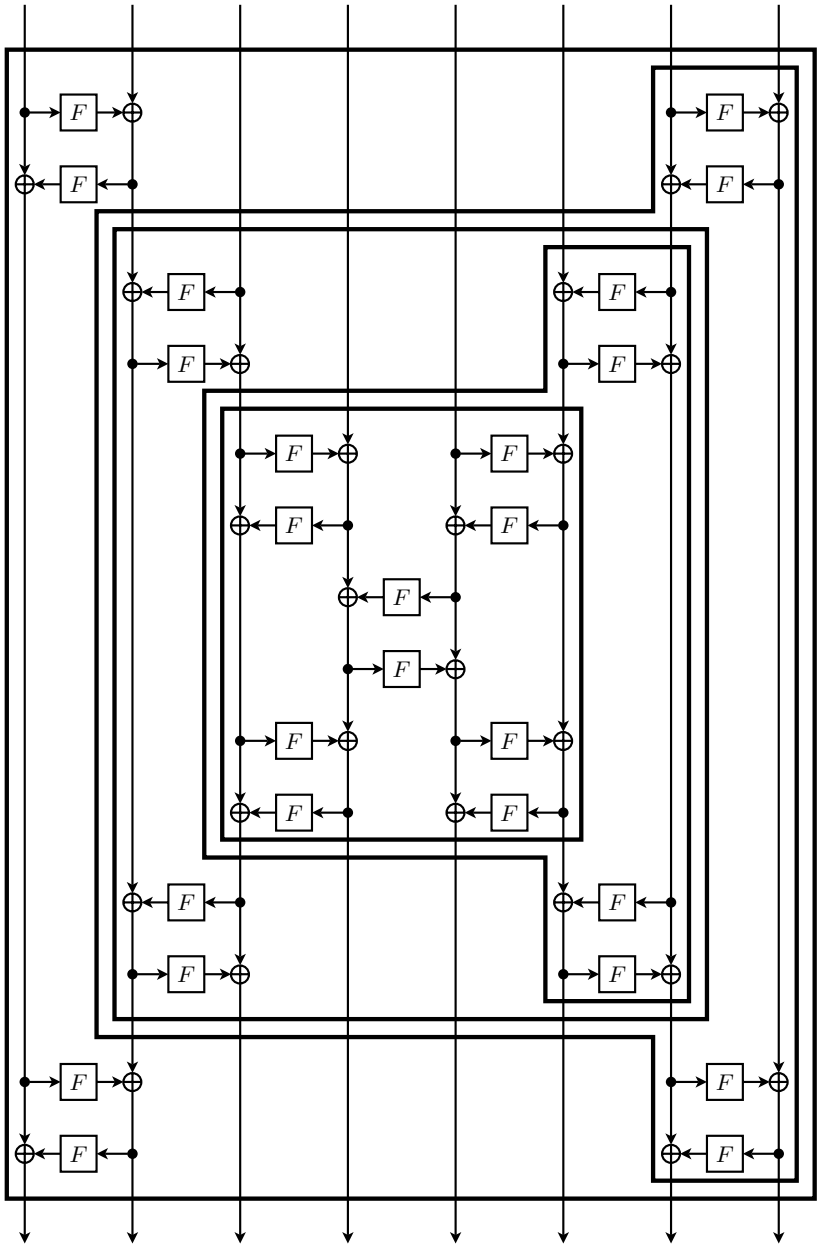


Fig. 3. The Simpira construction for $b \notin \{1, 2, 3, 4, 6, 8\}$. F is shorthand for $F_{c,b}$, which consists of two rounds of AES as specified in Alg. 2. A generic construction is shown for all $b \geq 4$, however for $b \in \{4, 6, 8\}$ we will use the construction of Fig. 2. By convention, the leftmost F-function is from left to right; when this is not the case in the diagram, the direction of every F-function should be inverted. The full-round Simpira iterates the construction in this diagram three times. See Sect. 4 for a full specification.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

construction, but also has the advantage ensuring at least 40 active S-boxes (instead of only 30) after five rounds.

But even if we had considered Yanagihara and Iwata's Type-1.x (b,2) GFS, we should also consider GFSs with more than two F-functions per Feistel round, which reach full subblock diffusion even quicker. However, this seems to come at the cost of using more F-functions in total. Looking only at the tabulated values of $\mathrm{DRmax}(\pi)$ and $\mathrm{DRmax}(\pi^{-1})$ in literature [75, 78–80], we can immediately rule out almost all alternative designs. Nevertheless, two improved Type-2 GFS designs by Suzaki and Minematsu [75] turned out to be superior. Instead of a cyclic left shift, they reorder the subblocks in a different way at the end of every Feistel round. We now explore these in detail.

Case $b = 6$. Let the *subblock shuffle* at the end of every Feistel round be presented by a list of indices that indicates which input subblock is mapped to which output subblock, e.g. $\{b-1, 0, 1, 2, \ldots, b-2\}$ denotes a cyclic left shift. Suzaki and Minematsu's improved Type-2 GFS with subblock shuffle $\{3, 0, 1, 4, 5, 2\}$ reaches full subblock diffusion and full bit diffusion after five Feistel rounds. At least 25 active S-boxes (in fact at least 30) are reached after four Feistel rounds. Following the design criteria, we end up with a design with 15 Feistel rounds. As this design has three F-functions in every Feistel round, it evaluates $3 \cdot 15 = 45$ F-functions. This is less than the general $b \geq 4$ case that requires $6b - 9$ Feistel rounds with 2 F-functions per round, which corresponds to $(6 \cdot 6 - 9) \cdot 2 = 54$ F-functions.

Case $b = 8$. Suzaki and Minematsu's improved Type-2 GFS with subblock shuffle $\{3, 0, 7, 4, 5, 6, 1, 2\}$ ensures both full subblock diffusion and full bit diffusion after six rounds. After four Feistel rounds, there are at least 25 active S-boxes (in fact at least 30). According to the design criteria, we end up with a design with 18 Feistel rounds, or $18 \cdot 4 = 72$ F-functions in total. The general $b \geq 4$ design would have required $(6b - 9) \cdot 2$ F-functions, which for $b = 8$ corresponds to $(6 \cdot 8 - 9) \cdot 2 = 78$ F-functions.

3.3 Design Alternatives

Until now, the only designs we discussed were GFS constructions where the F-function consists of two rounds of AES. We now take a step back, and briefly discuss alternative design choices.

As explained earlier, it is convenient to use two rounds of AES as a building block. It not only means that we reach full bit diffusion, but also that a "free XOR" is available to add a round constant on Intel and AMD architectures.

It is nevertheless possible to consider GFS designs with an F-function that consists of only one AES round. A consequence of this design choice is that extra XOR instructions will be needed to introduce round constants, which could increase the cycle count. But this design choice also complicates the analysis. For example when $b = 2$, we find that 25 Feistel rounds are then needed to ensure

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

at least 25 linearly active S-boxes. As shown in Fig. 4, this is because the tool can only ensure one active S-box for every Feistel round. Using two rounds of AES avoids this problem (see Fig. 5), and also significantly speeds up the tool: it makes bounding the minimum number of active S-boxes rather easy, instead of becoming increasingly complicated for a reasonably large value of b.

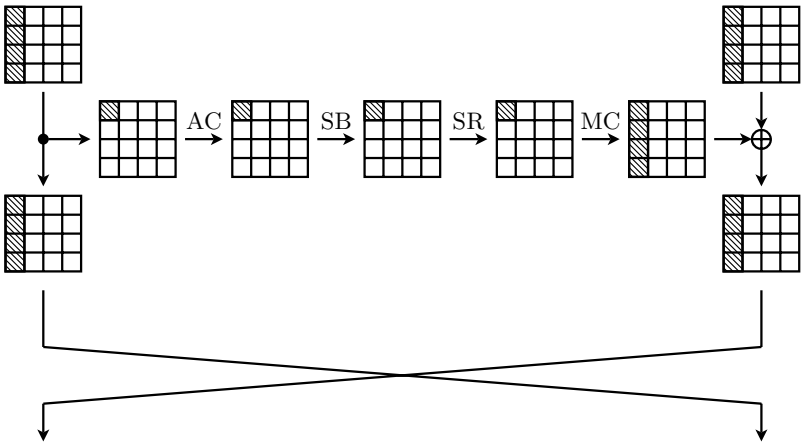


Fig. 4. A linear characteristic for an AES-based Feistel that uses only one round of AES inside its F-function. Crosshatches represent bytes with non-zero linear masks. The AES round consists of the `AddConstant` (AC), `SubBytes` (AC), `ShiftRows` (SR), and `MixColumns` (MC) operations. This round has only one active S-box. Therefore, 25 rounds are needed to ensure that there are least 25 linearly active S-boxes.

Likewise, we could also consider designs with more than two AES rounds per F-function. In our experiments, we have not found any cases where this results in a design where the total number of AES rounds is smaller. The intuition is as follows: the number of Feistel rounds to reach full subblock diffusion is independent of the F-function, therefore adding more AES rounds to every F function is not expected to result in a better trade-off.

If we take another step back, we might consider to use other instructions besides `AESENC`. Clearly, `AESDEC` can be used as an alternative, and the security properties and the benchmarks will remain the same. In fact, we use `AESDEC` when $b = 1$, to implement the inverse permutation. We do not use the `AESENCLAST` and `AESDECLAST` instructions, as they omit the `MixColumns` (resp. `InvMixColumns`) operation that is crucial to the *wide trail design strategy* [30] of AES. We do, however, use only one `AESENCLAST` for the very last round of the $b = 1$ permutation, as this makes an efficient implementation of the inverse permutation possible on Intel architectures. This is equivalent to applying a linear transformation to the output of the $b = 1$ permutation, therefore it does not reduce its cryptographic properties.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

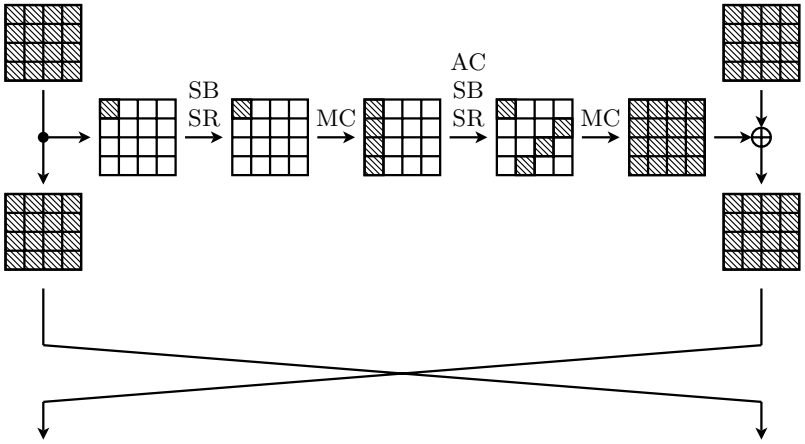


Fig. 5. A linear characteristic for one round of Simpira with $b = 2$ with 5 active S-boxes. Crosshatches represent bytes with non-zero linear masks. As Simpira uses two AES rounds per F-function, it can reach 25 active S-boxes in only 5 Feistel rounds, corresponding to 10 AES rounds in total.

Of course, it is possible to use non-AES instructions, possibly in combination with AES instructions. Actually, we do not need to be restricted to (generalized) Feistel designs for $b \geq 2$. However, such considerations are outside of the scope of this paper.

4 Specification of Simpira

An algorithmic specification of the Simpira design of Fig. 2–3 is given in Fig. 9–11. It uses one round of AES as a building block, which corresponds to the `AESENC` instruction on Intel processors (see Alg. 1). Its input is a 128-bit xmm register, which stores the AES 4×4 matrix of bytes as shown in Fig. 6. For additional details, we refer to [44].

s_0	s_4	s_8	s_{12}
s_1	s_5	s_9	s_{13}
s_2	s_6	s_{10}	s_{14}
s_3	s_7	s_{11}	s_{15}

Fig. 6. The internal state of AES can be represented by a 4×4 matrix of bytes, or as a 128-bit xmm register value $s = s_{15} \| \ldots \| s_0$, where s_0 is the least significant byte.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The F-function is specified in Alg. 2. It is parameterized by a counter c and by the number of subblocks b. Here, `SETR_EPI32` converts four 32-bit values into a 128-bit value, using the same byte ordering as the `_mm_setr_epi32()` compiler intrinsic. Fig. 7 shows how the constants can be expressed using the 4×4 byte matrix of AES.

$\texttt{0x00} \oplus c_0 \oplus b_0$	$\texttt{0x10} \oplus c_0 \oplus b_0$	$\texttt{0x20} \oplus c_0 \oplus b_0$	$\texttt{0x30} \oplus c_0 \oplus b_0$
$c_1 \oplus b_1$	$c_1 \oplus b_1$	$c_1 \oplus b_1$	$c_1 \oplus b_1$
$c_2 \oplus b_2$	$c_2 \oplus b_2$	$c_2 \oplus b_2$	$c_2 \oplus b_2$
$c_3 \oplus b_3$	$c_3 \oplus b_3$	$c_3 \oplus b_3$	$c_3 \oplus b_3$

Fig. 7. The constants used inside the $F_{c,b}$ function of Alg. 2, expressed as a 4×4 matrix of bytes. Here, $c = c_4 \| \ldots \| c_0$ and $b = b_4 \| \ldots \| b_0$ are 32-bit integers, where the least significant byte is c_0 and b_0 respectively.

Note that the constants have been updated in Simpira v2. The old constants of Simpira v1 are shown in Fig. 8. This update can be seen as "Grøstl strengthening," as it is inspired by the new round constants of the final-round Grøstl SHA-3 candidate [41]. No attack is currently known on Simpira v2 with the old rotation constants. Nevertheless, this change seems to strengthen Simpira without affecting its performance in our benchmarks.

c_0	b_0	0	0
c_1	b_1	0	0
c_2	b_2	0	0
c_3	b_3	0	0

Fig. 8. The old Simpira v1 constants used inside the $F_{c,b}$ function of Alg. 2, expressed as a 4×4 matrix of bytes. Again, $c = c_4 \| \ldots \| c_0$ and $b = b_4 \| \ldots \| b_0$ are 32-bit integers, where the least significant byte is x_0 and c_0 respectively.

Both the input and output of Simpira consist of b subblocks of 128 bits. The arrays use zero-based numbering, and array subscripts should be taken modulo the number of elements of the array. The subblock shuffle is done implicitly: we do not reorder the subblocks at the end of a Feistel round, but instead we apply the F-functions to other subblock inputs in the subsequent round. It is rather straightforward to implement the cyclic left shift in this way. For $b = 6$ and $b = 8$, the implementation of the subblock shuffle uses a decomposition into disjoint cycles.

As a result of this implementation choice, for $b \in \{2, 3, 4, 6, 8\}$, Simpira and its reduced-round variants are not always equivalent to a (generalized) Feistel

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

with identical rounds. For example, for $b = 2$ the F-function is alternatingly applied from left to right and from right to left. When the number of rounds is odd, this is not equivalent to a Feistel with identical rounds: the two output subblocks will be swapped.

When $b = 1$, an extra `InvMixColumns` operation is applied to the output. This is equivalent to omitting the `MixColumns` operation in the last round, and is required to efficiently implement the inverse Simpira permutation using Intel's AES instructions. For details on how to efficiently implement both Simpira and Simpira^{-1} when $b = 1$, see App. B.

The design strategy of Simpira is intended to be very conservative. Because we think that the security of Simpira with very large b may not yet be well understood, we recommend to use Simpira with $b \leq 65536$, corresponding to inputs of at most one megabyte. However, the external cryptanalysis of Simpira for any value of b is highly encouraged.

5 Benchmarks

We measured the performance of Simpira on the latest Intel processor, Architecture Codename Skylake. On this platform, the latency of `AESENC` is 4 cycles, and its throughput is 1 cycle. It follows that the software can be written in a way that fills the pipeline, by operating on four independent inputs. To obtain maximum throughput for all permutation sizes, we wrote functions that compute Simpira on four independent inputs. All Simpira permutations are benchmarked in the same setting, to make the results comparable.

Note that when $b = 4$, Simpira uses two independent F-functions, which means that maximum throughput could already be reached with only two independent inputs. For $b = 8$, where Simpira has four independent F-functions, even a single-stream Simpira implementation would fill the pipeline.

The measurements are performed as follows. We benchmark a function that evaluates Simpira for four independent inputs, and computed the number of cycles to carry out 256 calls to this function, as a "unit." This provides us with the throughput of Simpira. The results were obtained by using the `RDTSCP` instruction, 250 repetitions as a "warmup" phase, averaging the measurement on subsequent 1000 runs. Finally, this experiment was repeated 30 times, and the best result was selected. The platform was set up with Hyperthreading and Turbo Boost disabled.

The four data inputs can be stored sequentially at different pointers, or in an interleaved way (i.e. `A[0]B[0]C[0]D[0]A[1]B[1]C[1]D[1]...`). We benchmarked both settings. The results are shown in Table 1. We present only benchmarks for the forward Simpira permutation; the benchmarks for Simpira^{-1} turned out to be very similar.

We refer to App. D for a comparison with other constructions.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Algorithm 1 `AESENC` (see [44])

```
1: procedure AESENC(state, key)
2:     state ← SubBytes(state)
3:     state ← ShiftRows(state)
4:     state ← MixColumns(state)
5:     state ← state ⊕ key
6:     return state
7: end procedure
```

Algorithm 2 $F_{c,b}(x)$

```
1: procedure F_{c,b}(x)
2:     C ← SETR_EPI32(0x00 ⊕ c ⊕ b,
3:                    0x10 ⊕ c ⊕ b,
4:                    0x20 ⊕ c ⊕ b,
5:                    0x30 ⊕ c ⊕ b)
6:     return AESENC(AESENC(x, C), 0)
7: end procedure
```

Algorithm 3 Simpira ($b = 1$)

```
1: procedure SIMPIRA(x_0)
2:     R ← 6
3:     for c = 1, ..., R do
4:         x_0 ← F_{c,b}(x_0)
5:     end for
6:     InvMixColumns(x_0)
7:     return x_0
8: end procedure
```

Algorithm 4 Simpira^{-1} ($b = 1$)

```
1: procedure SIMPIRA(x_0)
2:     R ← 6
3:     MixColumns(x_0)
4:     for c = R, ..., 1 do
5:         x_0 ← F^{-1}_{c,b}(x_0)
6:     end for
7:     return x_0
8: end procedure
```

Algorithm 5 Simpira ($b \in \{2, 3, 4\}$)

```
1: procedure SIMPIRA(x_0, ..., x_{b-1})
2:     if (b = 2) ∨ (b = 3) then
3:         R ← 6b + 3
4:     else
5:         R ← 15
6:     end if
7:     c ← 1
8:
9:     for r = 0, ..., R − 1 do
10:        x_{r+1} ← x_{r+1} ⊕ F_{c,b}(x_r)
11:        c ← c + 1
12:        if b = 4 then
13:            x_{r+3} ← x_{r+3} ⊕ F_{c,b}(x_{r+2})
14:            c ← c + 1
15:        end if
16:    end for
17:    return (x_0, x_1, ..., x_{b-1})
18: end procedure
```

Algorithm 6 Simpira^{-1} ($b \in \{2, 3, 4\}$)

```
1: procedure SIMPIRA^{-1}(x_0, ..., x_{b-1})
2:     if (b = 2) ∨ (b = 3) then
3:         R ← 6b + 3
4:         c ← R
5:     else
6:         R ← 15
7:         c ← 2R
8:     end if
9:     for r = R − 1, ..., 0 do
10:        if b = 4 then
11:            x_{r+3} ← x_{r+3} ⊕ F_{c,b}(x_{r+2})
12:            c ← c − 1
13:        end if
14:        x_{r+1} ← x_{r+1} ⊕ F_{c,b}(x_r)
15:        c ← c − 1
16:    end for
17:    return (x_0, x_1, ..., x_{b-1})
18: end procedure
```

Fig. 9. Alg. 2 specifies $F_{c,b}$ using the `AESENC` operation that is defined in Alg. 1. Alg. 3–6 specify Simpira and its inverse for $b \leq 4$, where the input and output consist of b subblocks of 128 bits. Note that all arrays use zero-based numbering, and array subscripts should be taken modulo the number of elements of the array.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Algorithm 7 Simpira ($b = 6$)

procedure SIMPIRA($x_0, \ldots, x_5$)
  $R \leftarrow 15$
  $c \leftarrow 1$
  $s \leftarrow (0, 1, 2, 5, 4, 3)$
  for $r = 0, \ldots, R-1$ **do**
    $x_{s_{r+1}} \leftarrow x_{s_{r+1}} \oplus F_{c,b}(x_{s_r})$
    $c \leftarrow c + 1$
    $x_{s_{r+5}} \leftarrow x_{s_{r+5}} \oplus F_{c,b}(x_{s_{r+2}})$
    $c \leftarrow c + 1$
    $x_{s_{r+3}} \leftarrow x_{s_{r+3}} \oplus F_{c,b}(x_{s_{r+4}})$
    $c \leftarrow c + 1$
  end for
  return $(x_0, x_1, \ldots, x_5)$
end procedure

Algorithm 8 Simpira^{-1} ($b = 6$)

procedure SIMPIRA^{-1}($x_0, \ldots, x_5$)
  $R \leftarrow 15$
  $c \leftarrow 45$
  $s \leftarrow (0, 1, 2, 5, 4, 3)$
  for $r = R-1, \ldots, 0$ **do**
    $x_{s_{r+3}} \leftarrow x_{s_{r+3}} \oplus F_{c,b}(x_{s_{r+4}})$
    $c \leftarrow c - 1$
    $x_{s_{r+5}} \leftarrow x_{s_{r+5}} \oplus F_{c,b}(x_{s_{r+2}})$
    $c \leftarrow c - 1$
    $x_{s_{r+1}} \leftarrow x_{s_{r+1}} \oplus F_{c,b}(x_{s_r})$
    $c \leftarrow c - 1$
  end for
  return $(x_0, x_1, \ldots, x_5)$
end procedure

Algorithm 9 Simpira ($b = 8$)

procedure SIMPIRA($x_0, \ldots, x_7$)
  $R \leftarrow 18$
  $c \leftarrow 1$
  $s \leftarrow (0, 1, 6, 5, 4, 3)$
  $t \leftarrow (2, 7)$
  for $r = 0, \ldots, R-1$ **do**
    $x_{s_{r+1}} \leftarrow x_{s_{r+1}} \oplus F_{c,b}(x_{s_r})$
    $c \leftarrow c + 1$
    $x_{s_{r+5}} \leftarrow x_{s_{r+5}} \oplus F_{c,b}(x_{t_r})$
    $c \leftarrow c + 1$
    $x_{s_{r+3}} \leftarrow x_{s_{r+3}} \oplus F_{c,b}(x_{s_{r+4}})$
    $c \leftarrow c + 1$
    $x_{t_{r+1}} \leftarrow x_{t_{r+1}} \oplus F_{c,b}(x_{s_{r+2}})$
    $c \leftarrow c + 1$
  end for
  return $(x_0, x_1, \ldots, x_7)$
end procedure

Algorithm 10 Simpira^{-1} ($b = 8$)

procedure SIMPIRA^{-1}($x_0, \ldots, x_7$)
  $R \leftarrow 18$
  $c \leftarrow 72$
  $s \leftarrow (0, 1, 6, 5, 4, 3)$
  $t \leftarrow (2, 7)$
  for $r = R-1, \ldots, 0$ **do**
    $x_{t_{r+1}} \leftarrow x_{t_{r+1}} \oplus F_{c,b}(x_{s_{r+2}})$
    $c \leftarrow c - 1$
    $x_{s_{r+3}} \leftarrow x_{s_{r+3}} \oplus F_{c,b}(x_{s_{r+4}})$
    $c \leftarrow c - 1$
    $x_{s_{r+5}} \leftarrow x_{s_{r+5}} \oplus F_{c,b}(x_{t_r})$
    $c \leftarrow c - 1$
    $x_{s_{r+1}} \leftarrow x_{s_{r+1}} \oplus F_{c,b}(x_{s_r})$
    $c \leftarrow c - 1$
  end for
  return $(x_0, x_1, \ldots, x_7)$
end procedure

Fig. 10. Alg. 7–10 specify Simpira and its inverse for $b = 6$ and $b = 8$, using the $F_{c,b}$-function that is specified in Alg. 2. The input and the output consist of b subblocks of 128 bits. Note that all arrays use zero-based numbering, and array subscripts should be taken modulo the number of elements of the array.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Algorithm 11 Simpira ($b \notin \{1, 2, 3, 4, 6, 8\}$)

```
1: procedure SIMPIRA(x_0, ..., x_{b-1})
2:     k ← 0
3:     d ← 2 · ⌊b/2⌋
4:     for j = 1, ..., 3 do
5:         if d ≠ b then
6:             TWOF(b − 2, k)
7:             k ← k + 1
8:         end if
9:         for r = 0, ..., d − 2 do
10:            TWOF(r, k)
11:            k ← k + 1
12:            if r ≠ d − r − 2 then
13:                TWOF(d − r − 2, k)
14:                k ← k + 1
15:            end if
16:        end for
17:        if d ≠ b then
18:            TWOF(b − 2, k)
19:            k ← k + 1
20:        end if
21:    end for
22:    return (x_0, x_1, ..., x_{b-1})
23: end procedure

24: procedure TWOF(r, k)
25:    if r mod 2 = 0 then
26:        x_{r+1} ← x_{r+1} ⊕ F_{2k+1,b}(x_r)
27:        x_r ← x_r ⊕ F_{2k+2,b}(x_{r+1})
28:    else
29:        x_r ← x_r ⊕ F_{2k+1,b}(x_{r+1})
30:        x_{r+1} ← x_{r+1} ⊕ F_{2k+2,b}(x_r)
31:    end if
32: end procedure
```

Algorithm 12 Simpira^{-1} ($b \notin \{1, 2, 3, 4, 6, 8\}$)

```
1: procedure SIMPIRA^{-1}(x_0, ..., x_{b-1})
2:     k ← 6b − 10
3:     d ← 2 · ⌊b/2⌋
4:     for j = 1, ..., 3 do
5:         if d ≠ b then
6:             INVTWOF(b − 2, k)
7:             k ← k − 1
8:         end if
9:         for r = d − 2, ..., 0 do
10:            if r ≠ d − r − 2 then
11:                INVTWOF(d − r − 2, k)
12:                k ← k − 1
13:            end if
14:            INVTWOF(r, k)
15:            k ← k − 1
16:        end for
17:        if d ≠ b then
18:            INVTWOF(b − 2, k)
19:            k ← k − 1
20:        end if
21:    end for
22:    return (x_0, x_1, ..., x_{b-1})
23: end procedure

24: procedure INVTWOF(r, k)
25:    if r mod 2 = 0 then
26:        x_r ← x_r ⊕ F_{2k+2,b}(x_{r+1})
27:        x_{r+1} ← x_{r+1} ⊕ F_{2k+1,b}(x_r)
28:    else
29:        x_{r+1} ← x_{r+1} ⊕ F_{2k+2,b}(x_r)
30:        x_r ← x_r ⊕ F_{2k+1,b}(x_{r+1})
31:    end if
32: end procedure
```

Fig. 11. Alg. 11–12 specify Simpira and its inverse for $b \notin \{1, 2, 3, 4, 6, 8\}$, using the $F_{c,b}$-function that is specified in Alg. 2. Both the input and the output consist of b subblocks of 128 bits.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Table 1. Benchmarking results for the throughput of the Simpira permutations. For every b, we benchmark a function that applies the $128b$-bit permutation to four independent inputs. The data is either stored sequentially at different pointers, or in interleaved buffers. We give the number of cycles to process the four inputs, as well as the overhead compared the theoretical optimum of performing only `AESENC` instructions.

			non-interleaved		interleaved	
b	bits	# `AESENC`	cycles (4×)	overhead	cycles (4×)	overhead
1	128	12	50	3 %	50	3 %
2	256	30	122	1 %	122	1 %
3	384	42	171	2 %	171	2 %
4	512	60	241	1 %	241	1 %
6	768	90	362	1 %	362	1 %
8	1024	144	594	3 %	594	3 %
16	2048	348	1586	14 %	1400	1 %
32	4096	732	3295	13 %	2946	1 %
64	8192	1500	6791	13 %	6040	1 %
128	16384	3036	13942	15 %	12220	1 %
256	32768	6108	31444	29 %	24799	2 %

6 Cryptanalysis

The design criteria of Sect. 3 are not meant to be sufficient to guarantee security. In fact, it is not difficult to come up with trivially insecure constructions that satisfy (most of) the criteria. Rather, the design criteria are meant to assist us in identifying interesting constructions, which must then pass the scrutiny of cryptanalysis. Actually, during the design process of Simpira, we stumbled upon designs that were either insecure, or for which the security analysis was not so straightforward. When this happened, we adjusted the design criteria and repeated the search for constructions.

As such, we will not directly use the design criteria to argue the security of Simpira. Instead, we will use the fact that Simpira uses (generalized) Feistel structures and the AES round function, both of which have been extensively studied in literature. This allows us to focus our cryptanalysis efforts on the most promising attacks for this type of construction. We have tried to make this section easy to understand, which will hopefully convince the reader that Simpira should have a very comfortable security margin against all currently-known attacks.

Security claim. In what follows, we will only consider structural distinguishers [8] with a complexity up to 2^{128}. Simpira can be used in constructions that require a random permutation, however no statements can be made for adversaries that exceed 2^{128} queries. This type of security argument was first made by the SHA-3 [38] design team in response to high-complexity distinguishing

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

attacks on the underlying permutation [19–21], and has since been reused for other permutation-based designs.

Symmetry attacks. As explained in Sect. 3, the round constants are meant to avoid symmetry inside a Simpira round, as well as symmetry between rounds. The round constants also depend on b, which means that Simpira permutations of different widths should be indistinguishable from each other. The round constants are generated by a simple counter: this not only makes the design easy to understand and to implement, but also avoids any concerns that the constants may contain a backdoor. Every F-function has a different round constant: this does not seem to affect performance on recent Intel platforms, but greatly reduces the probability that a symmetry property can be maintained over several rounds.

Invariant subspace attacks. In its basic form, an invariant subspace attack [58] implies that there exists a coset of a vector space, so any number of iterations of the cryptographic round function maps to cosets of the same subspace. Rønjom [73] describes such an attack on Simpira v1 with $b = 4$, which is fixed in the current version. As explained in Sect. 9, no invariant subspace attacks were found for Simpira v2.

State collisions. For most block-cipher-based modes of operation, it is possible to define a "state," which is typically 128 bits long. This can be the chaining value for CBC mode, the counter for CTR mode, or the checksum in OCB. When a collision is found in this state, which is expected to happen around 2^{64} queries, the mode becomes insecure. For the Feistel-based Simpira ($b \geq 2$), there is no such concept of a "state." In fact: all subblocks receive roughly an equal amount of "processing." This allows Simpira to reach security beyond 2^{64} queries after a sufficient amount of Feistel rounds.

Linear and differential cryptanalysis. Simpira's security argument against linear [12] and differential [62] cryptanalysis (up to attacks with complexity 2^{128}) is the same as the argument for AES, which is based on counting the number of active S-boxes. As explained in [31], four rounds of AES have at least 25 (linearly or differentially) active S-boxes. Then any four-round differential characteristic holds with a probability less than $2^{-6 \cdot 25} = 2^{-150}$, and any four-round linear characteristic holds with a correlation less than $2^{-3 \cdot 25} = 2^{-75}$.

Here, 2^{-6} refers to the maximum difference propagation probability, and 2^{-3} is the maximum correlation amplitude of the S-box used in AES. The aforementioned reasoning makes the common assumptions that the probabilities of every round of a characteristic can be multiplied, and that this leads to a good estimate for the probability of the characteristic, and also of the corresponding differential.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The number of rounds typically needs to be slightly higher to account for partial key guesses (for keyed constructions), and to have a reasonable security margin. For any of the Simpira designs, we have at least three times the number of rounds required to reach 25 active S-boxes. This should give a sizable security margin against linear and differential cryptanalysis, and even against more advanced variants such as saturation and integral cryptanalysis [29]. In the case of integral cryptanalysis, of particular interest are the recently proposed integral distinguishers on Feistel and Generalized Feistel Networks by Todo [76] and by Zhang and Wenling [81].

Boomerang and differential-linear cryptanalysis. Instead of using one long characteristic, boomerang [77] and differential-linear [11, 57] cryptanalysis combine two shorter characteristics. But even combined with partial key guesses, the fact that Simpira has at least three times the number of rounds that result in 25 active S-boxes, should be more than sufficient to protect against this type of attacks.

Truncated and impossible differential cryptanalysis. When full bit diffusion is not reached, it is easy to construct a truncated differential [54] characteristic with probability one. A common way to construct an impossible differential [9,10] is the *miss in the middle* approach. It combines two probability-one truncated differentials, whose conditions cannot be met together.

However, every Simpira variant has at least three times the number of rounds to reach full bit diffusion. This should not only prevent truncated and impossible differential attacks, but result in a satisfactory security margin against such attacks.

Meet-in-the-middle and rebound attacks. Meet-in-the-middle-attacks [34] separate the equations that describe a symmetric-key primitive into two or three groups. This is done in such a way that some variables do not appear into at least one of these groups. A typical rebound attack [64] also splits a cipher into three parts: an inner part that is satisfied by meet-in-the-middle techniques (in the inbound phase), and two outer parts that are fulfilled in a probabilistic way (in the outbound phase).

With Simpira, splitting the construction in three parts will always result in one part that either has at least 25 active S-boxes, or that reaches full bit diffusion. This should not only prevent meet-in-the-middle and rebound attacks, but also provide a large security margin against these attacks.

On Simpira with $b = 1$ (corresponding to 12-round AES with fixed round keys), the best known distinguisher is a rebound attack by Gilbert and Peyrin [42] that attacks 8 rounds out of 12.

Generic attacks. A substantial amount of literature exists on generic attacks of Feistel structures. In particular, we are interested in attacks in Maurer et

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

al.'s indifferentiability setting [63], which is an extension of the indistinguishability notion for constructions that use publicly available oracles. In Simpira, the F-functions contain no secret key, and are therefore assumed to be publicly available.

Coron et al. [25] showed that five rounds of Feistel are not indifferentiable from a random permutation, and presented a indifferentiability proof for six rounds. Holenstein et al. [50] later showed that their proof is flawed, and provided a new indifferentiability proof for fourteen rounds. In very recent work, Dai and Steinberger [32] and independently Dachman-Soled et al. [28] announced an indifferentiability proof for the 10-round Feistel, which Dai and Steinberger subsequently improved to a proof for 8 rounds [33].

A problem with the aforementioned indifferentiability proofs is that they are rather weak: if the F-function is 128 bits wide, security is only proven up to about 2^{16} queries. The indistinguishability setting is better understood, where many proofs are available for not only Feistel, but also various generalized Feistel structures. But even in this setting, most proofs do not go beyond 2^{64} queries, and proving security with close to 2^{128} queries requires a very large number of rounds [49].

So although several of Simpira's Feistel-based permutations were proven to be indistinguishable from random permutations using [65,82], it is an open problem to prove stronger security bounds for Simpira and other generalized Feistel structures. Nevertheless, no generic attacks are known for Simpira, even when up to 2^{128} are made.

Note that strictly speaking, there is an exception to the previous sentence for Simpira with $b = 1$. It is guaranteed to be an even permutation [22, Thm. 4.8], and therefore $2^{128} - 1$ queries can distinguish it from a random permutation with advantage 0.5. We only mention this for completeness; actually all of Simpira's permutations can be shown to be even, but this is typically not considered to be more than just a mathematical curiosity.

Other attacks. We do not consider brute-force-like attacks [70], such as the biclique attacks on AES [16]: they perform exhaustive search on a smaller number of rounds, and therefore do not threaten the practical security of the cipher. However, it will be interesting to investigate such attacks in future work, as they give an indication of the security of the cipher in the absence of other attacks. We also do not look into algebraic attacks, as AES seems to very resistant against such attacks.

7 Applications

Simpira can be used in various scenarios where AES does not permit an efficient construction with security up to 2^{128} evaluations of the permutation. We present a brief overview possible applications.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

A block cipher without round keys. The (single-key) Even-Mansour construction [37, 39, 40] uses a secret key K to turn a plaintext P into a ciphertext C as follows:

$$C = E_K(P) = \pi(P \oplus K) \oplus K \ , \tag{1}$$

where π is an n-bit permutation. As argued by Dunkelman et al. [37], the construction is minimal, in the sense that simplifying it, for example by removing one of its components, will render it completely insecure. Mouha and Luykx [67] showed that the Even-Mansour is in some sense optimal in the multi-key setting, where several keys are independently and uniformly drawn from the key space.

When D plaintext-ciphertexts are available, the secret key K of the Even-Mansour construction can be recovered in $2^n/D$ (off-line) evaluations of the permutation π [37]. This may be acceptable in lightweight authentication algorithms which rekey regularly, but may not be sufficient for encryption purposes [66, 67]. In order to achieve security up to about 2^{128} queries against all attacks in the multi-key setting, the Even-Mansour construction requires a permutation of at least 256 bits.

An important advantage of the Even-Mansour construction is that it avoids the need to precalculate round keys (and store them securely!) or to calculate them on the fly. Moreover, it also allows the easy construction of a tweakable block cipher. For a given tweak T, one can turn the Even-Mansour construction into a tweakable block cipher [59, 60]:

$$C = E_K(P) = \pi(P \oplus K \cdot T) \oplus K \cdot T \ , \tag{2}$$

that can be proven to be secure up to $2^{n/2}$ queries in the multi-key setting using the proof of [67,68]. For concreteness, we use the multiplication $K \cdot T$ in $GF(2^n)$, which restricts the tweaks to $T \neq 0$. However, any ϵ-AXU hash function can be used instead of this multiplication [23].

If the cipher is computed in a parallelizable mode of operation, independent blocks can be pipelined, and the performance would be dominated by Simpira with the relevant value of b, plus the overhead of the key addition.

Permutation-based hashing. Achieving 128-bit collision resistance with a 128-bit permutation has been shown to be impossible [71]. Typically, a large permutation size is used to achieve a high throughput, for example 1600 bits in the sponge construction of SHA-3 [38]. The downside of using a large permutation is that performance is significantly reduced when many short messages need to be hashed, for example to compute a Lamport signature [56]. Simpira overcomes these problems by providing a family of efficient permutations with different input sizes.

In particular for hashing short messages, one may consider to use Simpira with a Davies-Meyer feed-forward: $\pi(x) \oplus x$. This construction has been shown to be optimally preimage and collision-resistant [14, 15], and even preimage aware [36], but not indifferentiable from a random oracle [24] as it is easy to find a fixed point: $\pi^{-1}(0)$. To match the intended application, padding of the input and/or truncation of the output of Simpira may be required.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Wide-block encryption and robust authenticated encryption. Wide-block encryption can be used to provide security against chosen ciphertext attacks when short (or even zero-length) authentication tags are used. In the context of full-disk encryption, there is usually no space to store an authentication tag. In an attempt to reduce the risk that ciphertext changes result in meaningful plaintext, a possibility is to use a wide block cipher to encrypt an entire disk sector, which typically has a size of 512 to 4096 bytes.

The same concern also exists when short authentication tags are used, and can be addressed by an encode-then-encipher approach [5]: add some bits of redundancy, and then encrypt with an arbitrary-input-length block cipher. Note that this technique achieves robust authenticated encryption [48].

Typical solutions for wide-block encryption such as the VIL [4], CMC [46] and EME [45, 47] modes of operation have the disadvantage that they are patented, and do not provide security beyond 2^{64} blocks of input. We are unaware of any patents related to Simpira.

When used in an Even-Mansour construction, Simpira with $b \geq 2$ can provide a wide block cipher that provides security up to 2^{128} blocks. When the block size exceeds the key size, the Even-Mansour construction can be generalized as follows:

$$C = E_K(P) = \pi(P \oplus (K \cdot T) \| 0^*) \oplus (K \cdot T) \| 0^*) \ , \tag{3}$$

where we set $T = 1$ if no tweak is provided. Note that this straightforward extension of the Even-Mansour construction appears in the proof for various sponge constructions. The first proof of security of this construction in the multi-key setting was given by Andreeva et al. [3].

8 A Problem with Yanagihara and Iwata's GFS

For $b \geq 4$ (except $b = 6$ and $b = 8$), Simpira v1 used Yanagihara and Iwata's Type-1.x (b,2) GFS [80]. This is a GFS with two F-functions per round, shown in Fig. 12. Strictly speaking, we consider a variant of Yanagihara and Iwata's construction, that is identical up to a reordering of the input and output sub-blocks.

This construction has a problem. As can be seen from Fig. 12, the same value x_0 will eventually be processed by two F-functions. This clearly results in a redundant calculation, as the same F-function is evaluated twice on the same input.

In Simpira v1, the F-functions are not entirely identical due to the round constants. However, it can be seen that the problem in Yanagihara and Iwata's Type-1.x GFS also results in an attack on Simpira v1. In particular, the bounds on the number of active S-boxes were not correct, as the exact same S-box transitions were counted more than once. Dobraunig et al. [35] exploited the fact that the actual number of active S-boxes is much lower than expected, and constructed a series of attacks on the full 15-round Simpira v1 with $b = 4$, including a collision attack with complexity $2^{82.62}$ on Simpira when it is used in a truncated Davies-Meyer hash construction.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

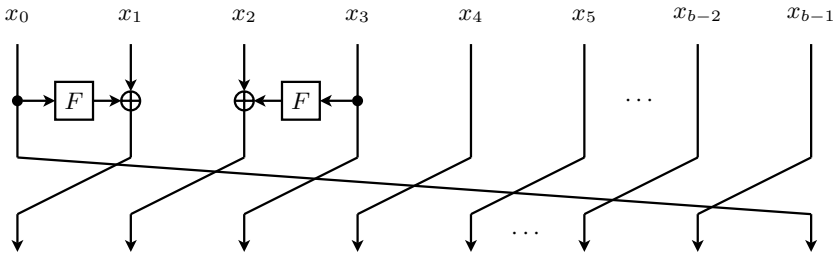


Fig. 12. Yanagihara and Iwata's Type-1.x (b,2) GFS [80], which was used in Simpira v1. Note that regardless of b, the same x_0 will eventually enter an F-function twice after a sufficient number of rounds.

The problem with Yanagihara and Iwata's construction was confirmed to us by its designers. It was pointed out to us that their Type-1.x GFS was implicitly assumed to use independent round keys, but that this assumption was unfortunately not mentioned in their paper [80].

When this assumption does not hold, the counts of active S-boxes can be incorrect. This occurs when various simple key schedules are used, such as for example the Even-Mansour construction. We avoid this problem in Simpira v2 by ensuring that the same input is never processed by more than one F-function. This can be seen to avoid attacks on GFS in block-cipher-based constructions, when used with a uniformly random key.

But Simpira is designed to be a family of cryptographic permutations, and should therefore also be secure in unkeyed settings. In the next section, we show how the unkeyed setting leads to invariant subspace attacks on Simpira v1 for $b = 4$.

9 Invariant Subspace Attacks

Leander et al. [58] introduced the term *invariant subspace attack*, which applies when there exists a (large) subspace, so that any coset of this subspace is mapped to itself when the round function is applied. We now explain such an attack applies to Yanagihara and Iwata Type-1.x (4,2) GFS. Again, strictly speaking Yanagihara and Iwata defined a variant of this construction, that is, however, identical up to a reordering of the input and output blocks. As illustrated in Fig. 13, we find that if the second and the last subblock of the input are identical, this property is preserved after any multiple of two rounds.

A similar observation also holds for Simpira v1 with $b = 4$, where only the round constants slightly destroy the symmetry property of the input. This is a consequence of the sparse round constants in Simpira v1, and the reuse of values into several F-functions, as explained in Sect. 8. In particular, for any even multiple of rounds up to 126, Simpira v1 round constants (see Fig. 8) only differ in the zeroth byte of the AES state. This means that if the second and the

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

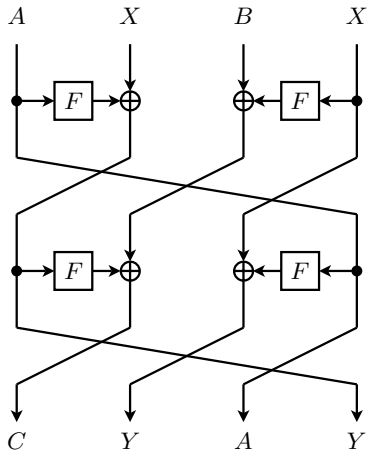


Fig. 13. Yanagihara and Iwata Type-1.x (4,2) GFS [80], which was used in Simpira v1 for $b = 4$. We assume that all F-functions are identical. Here A, B and X can be any value. The leftmost input subblock enters the same F-function twice, and therefore guarantees that the output value Y will appear twice as well.

last subblock of the input are identical, this property will be preserved, except for the first column of corresponding AES states.

Rønjom [73] described an invariant subspace attack on Simpira v1 with $b = 4$. In particular, Rønjom identified a large subspace such that any coset of this space is invariant under two rounds of Simpira v2. This leads to a plaintext invariant over infinitely many even rounds. It can be seen as a generalization of the attack on Yanagihara-Iwata's Type-1.x GFS that is described in this section.

The property does not hold for an odd number of rounds, so it does not apply directly for Simpira v1 with $b = 4$, which consists of 15 Feistel rounds. For this reason, we did not detect any non-randomness in our test vectors, although it included the all-zero input that is an element of the coset of the invariant subspace. However, simply applying the permutation twice means that the total number of rounds is even, so that the distinguisher applies.

Do such invariant subspaces attacks also exist for Simpira v2? In an attempt to find such attacks, we first look for invariant subspaces when all F-functions are identical. This should give a good starting point to find invariant subspaces when the real (non-identical) F-functions of Simpira are used. More specifically, we select a random F-function, and consider four values for every input subblock: 0, $F(0)$, $F(F(0))$ and $F(F(0)) \oplus F(0)$. We then apply the Feistel round function several times, and use Gaussian elimination to check whether we stay within a particular linear subspace.

Using this technique, we found invariant subspaces for the GFS used in Simpira v2 when $b \in \{4, 6, 8\}$ (i.e. assuming identical F-functions), but not for other values of b. In fact, it can be seen that there is an invariant subspace for any

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Type-2 GFS with an "even-odd shuffle [75]," that is, where even-numbered input subblocks are mapped to odd-numbered output subblocks and vice versa. For $b = 4$, such an invariant subspace is shown in Fig. 14. With the introduction of appropriate round constants, however, these invariant subspace attacks are avoided.

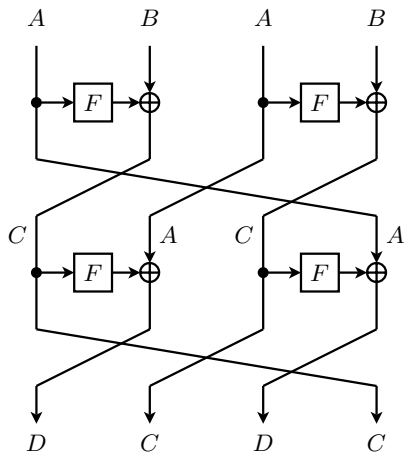


Fig. 14. The Type-2 GFS with $b = 4$, used in Simpira v2. We assume that all F-functions are identical. Here, A and B can be any value. If the odd-numbered input subblocks are equal, and the even-numbered input subblocks are equal, then this property is preserved for any number of rounds.

We chose to retain Type-2 GFS in Simpira v2 for $b \in \{4, 6, 8\}$, instead of replacing them by Generalized Feistel structures that "inherently" avoid invariant subspace attacks. This is because Type-2 GFS constructions are efficient and well-analyzed, and invariant subspaces can be avoided by using round constants.

We searched for invariant subspaces in all Simpira v2 variants, but were unable to find any. A similar search was also performed by Rønjom [72], who also could not identify invariant subspaces in the updated Simpira design. Unfortunately, currently no provable arguments against invariant subspace attacks are known. This is an interesting topic for future work.

10 Conclusion

We introduced Simpira, which is a family of cryptographic permutations that processes inputs of $128 \times b$ bits. It is intended to be a very conservative design that achieves high throughput on processors with AES instructions. We decided to use two rounds of AES as a building block, with the goal of simplifying

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

the design space exploration, and making the cryptanalysis and implementation straightforward.

With this building block, we explored a large number of generalized Feistel structures, and calculated how many rounds are required to reach either full bit diffusion, or 25 linearly or differentially active S-boxes, whichever is greater. To ensure a large security margin, we multiplied this number of rounds by three. Of all designs that we considered, we selected the ones with the lowest amount of F-functions in total.

Following these design criteria, Simpira resulted in seven different designs. For $b = 1$, we have AES with fixed round keys. Simpira uses a Feistel structure for $b = 2$, a Type-1 GFS for $b = 3$, and a Type-2 GFS for $b = 4$. The $b \geq 5$ design is a dedicated construction that we introduce in this paper. For $b = 6$ and $b = 8$, we use Suzaki and Minematsu's improved Type-2 GFS, as it has fewer F-functions than general construction for $b \geq 5$.

Our benchmarks on Intel Skylake showed that Simpira is close to the theoretical optimum of only executing `AESENC` instructions. For $b \leq 4$, Simpira is less than 3 % away from this optimum. For $b \leq 32$, corresponding to inputs of up to 512 bytes, Simpira is less than 13 % away from this optimum for a non-interleaved implementation, and less than 1 % away for an interleaved implementation.

It is unfortunate that many methods to encrypt wide input blocks, such as VIL, CMC, and EME, have not seen widespread adoption. The main obstacle appears to be that they are patented. We hope that Simpira can provide an interesting alternative: it is not only free from patent concerns, but offers security way beyond the 2^{64} limit for typical AES-based modes.

Acknowledgments. We thank the organizers and participants of Dagstuhl Seminar 16021, where an early version of this work was presented. The detailed comments and suggestions of the seminar participants helped to improve this manuscript significantly. Thanks to Christoph Dobraunig, Maria Eichlseder, Florian Mendel and Sondre Rønjom their attacks on Simpira v1, which lead to the updated Simpira v2 that is presented in this document. We also thank Eik List for pointing out some notation issues in an earlier version of this text, and Sébastien Duval, Brice Minaud, Kazuhiko Minematsu, and Tetsu Iwata for their insights into Feistel structures. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007), by Research Fund KU Leuven, OT/13/071, by the PQCRYPTO project, which was partially funded by the European Commission Horizon 2020 research Programme, grant #645622, by the ISRAEL SCIENCE FOUNDATION (grant No. 1018/16), and by the French Agence Nationale de la Recherche through the BLOC project under Contract ANR-11-INS-011, and the BRUTUS project under Contract ANR-14-CE28-0015. Nicky Mouha is supported by a Postdoctoral Fellowship from the Flemish Research Foundation (FWO-Vlaanderen), and by FWO travel grant 12F9714N. Certain algorithms and commercial products are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the algorithms or products identified are necessarily the best available for the purpose.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

References

1. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 430–454. Springer (2015)
2. Anderson, R.J., Biham, E.: Two Practical and Provably Secure Block Ciphers: BEARS and LION. In: FSE 1996. LNCS, vol. 1039, pp. 113–120. Springer (1996)
3. Andreeva, E., Daemen, J., Mennink, B., Assche, G.V.: Security of Keyed Sponge Constructions Using a Modular Proof Approach. In: FSE 2015. LNCS, vol. 9054, pp. 364–384. Springer (2015)
4. Bellare, M., Rogaway, P.: On the Construction of Variable-Input-Length Ciphers. In: FSE 1999. LNCS, vol. 1039, pp. 231–244. Springer (1999)
5. Bellare, M., Rogaway, P.: Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In: ASIACRYPT 2000. LNCS, vol. 1976, pp. 317–330. Springer (2000)
6. Berger, T.P., Francq, J., Minier, M., Thomas, G.: Extended Generalized Feistel Networks Using Matrix Representation to Propose a New Lightweight Block Cipher: Lilliput. IEEE Trans. Computers 65(7), 2074–2089 (2016)
7. Berger, T.P., Minier, M., Thomas, G.: Extended Generalized Feistel Networks Using Matrix Representation. In: SAC 2013. LNCS, vol. 8282, pp. 289–305. Springer (2013)
8. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Cryptographic Sponge Functions, available at `http://sponge.noekeon.org/CSF-0.1.pdf`
9. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: EUROCRYPT 1999. LNCS, vol. 1592, pp. 12–23. Springer (1999)
10. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. J. Cryptology 18(4), 291–311 (2005)
11. Biham, E., Dunkelman, O., Keller, N.: Enhancing Differential-Linear Cryptanalysis. In: ASIACRYPT 2002. LNCS, vol. 2501, pp. 254–266. Springer (2002)
12. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. J. Cryptology 4(1), 3–72 (1991)
13. Biryukov, A., Khovratovich, D.: PAEQ: Parallelizable Permutation-Based Authenticated Encryption. In: ISC 2014. LNCS, vol. 8783, pp. 72–89. Springer (2014)
14. Black, J., Rogaway, P., Shrimpton, T.: Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In: CRYPTO 2002. LNCS, vol. 2442, pp. 320–335. Springer (2002)
15. Black, J., Rogaway, P., Shrimpton, T., Stam, M.: An Analysis of the Blockcipher-Based Hash Functions from PGV. J. Cryptology 23(4), 519–545 (2010)
16. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. In: ASIACRYPT 2011. LNCS, vol. 7073. Springer (2011)
17. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., Verbauwhede, I.: SPONGENT: A Lightweight Hash Function. In: CHES 2011. LNCS, vol. 6917, pp. 312–325. Springer (2011)
18. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., Verbauwhede, I.: SPONGENT: The Design Space of Lightweight Cryptographic Hashing. IEEE Trans. Computers 62(10), 2041–2053 (2013)
19. Boura, C., Canteaut, A.: A zero-sum property for the KECCAK-f permutation with 18 rounds. In: ISIT 2010. pp. 2488–2492. IEEE (2010)

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

20. Boura, C., Canteaut, A.: Zero-Sum Distinguishers for Iterated Permutations and Application to Keccak-f and Hamsi-256. In: SAC 2010. LNCS, vol. 6544, pp. 1–17. Springer (2011)
21. Boura, C., Canteaut, A., De Cannière, C.: Higher-order differential properties of Keccak and Luffa. Cryptology ePrint Archive, Report 2010/589 (2010)
22. Cid, C., Murphy, S., Robshaw, M.J.B.: Algebraic Aspects of the Advanced Encryption Standard. Springer (2006)
23. Cogliati, B., Lampe, R., Seurin, Y.: Tweaking Even-Mansour Ciphers. In: CRYPTO 2015. LNCS, vol. 9215, pp. 189–208. Springer (2015)
24. Coron, J., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function. In: CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer (2005)
25. Coron, J., Patarin, J., Seurin, Y.: The Random Oracle Model and the Ideal Cipher Model Are Equivalent. In: CRYPTO 2008. LNCS, vol. 5157, pp. 1–20. Springer (2008)
26. Cossíos, D.: Breve Bestiario Peruano. Editorial Casatomada, second edn. (2008)
27. Crowley, P.: Mercy: A Fast Large Block Cipher for Disk Sector Encryption. In: FSE 2000. LNCS, vol. 1978, pp. 49–63. Springer (2000)
28. Dachman-Soled, D., Katz, J., Thiruvengadam, A.: 10-Round Feistel is Indifferentiable from an Ideal Cipher. In: EUROCRYPT 2016. LNCS, vol. 9666, pp. 649–678. Springer (2016)
29. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher Square. In: FSE 1997. LNCS, vol. 1267, pp. 149–165. Springer (1997)
30. Daemen, J., Rijmen, V.: The Wide Trail Design Strategy. In: IMA 2001. LNCS, vol. 2260, pp. 222–238. Springer (2001)
31. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
32. Dai, Y., Steinberger, J.: Indifferentiability of 10-Round Feistel Networks. Cryptology ePrint Archive, Report 2015/874 (2015)
33. Dai, Y., Steinberger, J.P.: Indifferentiability of 8-Round Feistel Networks. In: CRYPTO 2016. LNCS, vol. 9814, pp. 95–120. Springer (2016)
34. Diffie, W., Hellman, M.E.: Exhaustive Cryptanalysis of the NBS Data Encryption Standard. Computer 10(6), 74–84 (1977)
35. Dobraunig, C., Eichlseder, M., Mendel, F.: Cryptanalysis of Simpira. Cryptology ePrint Archive, Report 2016/244 (2016)
36. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging Merkle-Damgård for Practical Applications. In: EUROCRYPT 2009. LNCS, vol. 5479, pp. 371–388. Springer (2009)
37. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in Cryptography: The Even-Mansour Scheme Revisited. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 336–354. Springer (2012)
38. Dworkin, M.J.: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Federal Inf. Process. Stds. (NIST FIPS) - 202 (August 2015)
39. Even, S., Mansour, Y.: A Construction of a Cipher From a Single Pseudorandom Permutation. In: ASIACRYPT 1991. LNCS, vol. 739. Springer (1993)
40. Even, S., Mansour, Y.: A Construction of a Cipher from a Single Pseudorandom Permutation. J. Cryptology 10(3), 151–162 (1997)
41. Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: Grøstl – a SHA-3 candidate. Submission to the NIST SHA-3 Competition (Round 3) (2011), `http://www.groestl.info/Groestl.pdf`

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

42. Gilbert, H., Peyrin, T.: Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In: FSE 2010. LNCS, vol. 6147, pp. 365–383. Springer (2010)
43. Gueron, S.: Intel's New AES Instructions for Enhanced Performance and Security. In: FSE 2009. LNCS, vol. 5665, pp. 51–66. Springer (2009)
44. Gueron, S.: Intel® Advanced Encryption Standard (AES) New Instructions Set. Available at: `https://software.intel.com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set` (September 2012), Revision 3.01
45. Halevi, S.: EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data. In: INDOCRYPT 2004. LNCS, vol. 3348, pp. 315–327. Springer (2004)
46. Halevi, S., Rogaway, P.: A Tweakable Enciphering Mode. In: CRYPTO 2003. LNCS, vol. 2729, pp. 482–499. Springer (2003)
47. Halevi, S., Rogaway, P.: A Parallelizable Enciphering Mode. In: CT-RSA 2004. LNCS, vol. 2964, pp. 292–304. Springer (2004)
48. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust Authenticated-Encryption AEZ and the Problem That It Solves. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 15–44. Springer (2015)
49. Hoang, V.T., Rogaway, P.: On Generalized Feistel Networks. In: CRYPTO 2010. LNCS, vol. 6223, pp. 613–630. Springer (2010)
50. Holenstein, T., Künzler, R., Tessaro, S.: The equivalence of the random oracle model and the ideal cipher model, revisited. In: STOC 2011. pp. 89–98. ACM (2011)
51. Jean, J.: Cryptanalysis of Haraka. Cryptology ePrint Archive, Report 2016/396 (2016)
52. Jean, J., Nikolić, I., Sasaki, Y., Wang, L.: Practical Cryptanalysis of PAES. In: SAC 2014. LNCS, vol. 8781, pp. 228–242. Springer (2014)
53. Jean, J., Nikolić, I., Sasaki, Y., Wang, L.: Practical Forgeries and Distinguishers against PAES. IEICE Transactions 99-A(1), 39–48 (2016)
54. Knudsen, L.R.: Truncated and Higher Order Differentials. In: FSE 1994. LNCS, vol. 1008, pp. 196–211. Springer (1994)
55. Kölbl, S., Lauridsen, M.M., Mendel, F., Rechberger, C.: Haraka - Efficient Short-Input Hashing for Post-Quantum Applications. Cryptology ePrint Archive, Report 2016/098 (2016)
56. Lamport, L.: Constructing Digital Signatures from a One Way Function. Tech. Rep. SRI-CSL-98, SRI International Computer Science Laboratory (October 1979)
57. Langford, S.K., Hellman, M.E.: Differential-Linear Cryptanalysis. In: CRYPTO 1994. LNCS, vol. 839, pp. 17–25. Springer (1994)
58. Leander, G., Abdelraheem, M.A., AlKhzaimi, H., Zenner, E.: A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In: CRYPTO 2011. LNCS, vol. 6841, pp. 206–221. Springer (2011)
59. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable Block Ciphers. In: CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer (2002)
60. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable Block Ciphers. J. Cryptology 24(3), 588–613 (2011)
61. Lucks, S.: BEAST: A Fast Block Cipher for Arbitrary Blocksizes. In: CMS 1996. IFIP Conference Proceedings, vol. 70, pp. 144–153. Chapman & Hall (1996)
62. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer (1994)

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

63. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer (2004)
64. Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl. In: FSE 2009. LNCS, vol. 5665, pp. 260–276. Springer (2009)
65. Moriai, S., Vaudenay, S.: On the Pseudorandomness of Top-Level Schemes of Block Ciphers. In: ASIACRYPT 2000. LNCS, vol. 1976, pp. 289–302. Springer (2000)
66. Mouha, N.: The Design Space of Lightweight Cryptography. Cryptology ePrint Archive, Report 2015/303 (2015)
67. Mouha, N., Luykx, A.: Multi-key Security: The Even-Mansour Construction Revisited. In: CRYPTO 2015. LNCS, vol. 9215, pp. 209–223. Springer (2015)
68. Mouha, N., Mennink, B., Herrewege, A.V., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers. In: SAC 2014. LNCS, vol. 8781, pp. 306–323. Springer (2014)
69. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In: Inscrypt 2011. LNCS, vol. 7537, pp. 57–76. Springer (2011)
70. Rechberger, C.: On Bruteforce-Like Cryptanalysis: New Meet-in-the-Middle Attacks in Symmetric Cryptanalysis. In: ICISC 2012. LNCS, vol. 7839, pp. 33–36. Springer (2013)
71. Rogaway, P., Steinberger, J.P.: Security/Efficiency Tradeoffs for Permutation-Based Hashing. In: EUROCRYPT 2008. LNCS, vol. 4965, pp. 220–236. Springer (2008)
72. Rønjom, S.: Personal Communication (March 2016)
73. Rønjom, S.: Invariant subspaces in Simpira. Cryptology ePrint Archive, Report 2016/248 (2016)
74. Schroeppel, R.: The Hasty Pudding Cipher – A Tasty Morsel (1998), submission to the NIST AES competition
75. Suzaki, T., Minematsu, K.: Improving the Generalized Feistel. In: FSE 2010. LNCS, vol. 6147, pp. 19–39. Springer (2010)
76. Todo, Y.: Structural Evaluation by Generalized Integral Property. In: EUROCRYPT 2015. LNCS, vol. 9056. Springer (2015)
77. Wagner, D.: The Boomerang Attack. In: FSE 1999. LNCS, vol. 1636, pp. 156–170. Springer (1999)
78. Yanagihara, S., Iwata, T.: On Permutation Layer of Type 1, Source-Heavy, and Target-Heavy Generalized Feistel Structures. In: CANS 2011. LNCS, vol. 7092, pp. 98–117. Springer (2011)
79. Yanagihara, S., Iwata, T.: Improving the Permutation Layer of Type 1, Type 3, Source-Heavy, and Target-Heavy Generalized Feistel Structures. IEICE Transactions 96-A(1), 2–14 (2013)
80. Yanagihara, S., Iwata, T.: Type 1.x Generalized Feistel Structures. IEICE Transactions 97-A(4), 952–963 (2014)
81. Zhang, H., Wu, W.: Structural Evaluation for Generalized Feistel Structures and Applications to LBlock and TWINE. In: INDOCRYPT 2015. LNCS, vol. 9462, pp. 218–237. Springer (2015)
82. Zheng, Y., Matsumoto, T., Imai, H.: On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses. In: CRYPTO 1989. LNCS, vol. 435, pp. 461–480. Springer (1990)

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

A Simpira for $b \notin \{1, 2, 3, 4, 6, 8\}$: Full Bit Diffusion and Active S-Boxes

For the Simpira construction with $b \notin \{1, 2, 3, 4, 6, 8\}$, it is very straightforward to show that $4b-6$ F-functions are sufficient to reach full bit diffusion, as well as to ensure that at least 30 S-boxes are active. This can be proven by induction.

Full Bit Diffusion. Recall that full bit diffusion means that every output bit depends on every input bit, and every input bit depends on every output bit. Observe that the construction of Fig. 3 reaches full bit diffusion for $b = 4$. From the same figure, it can also easily be seen that the four additional F-functions ensure that full bit diffusion is reached for $b + 1$, provided that full bit diffusion was already reached for b.

Active S-boxes. The MILP tool shows us that the construction of Fig. 3 reaches at least 30 (linearly or differentially) active S-boxes for $b = 4$. We now prove that if the construction for b has at least 30 active S-boxes, then it has at least 30 active S-boxes for $b+1$. When a non-zero difference or a non-zero linear mask enters into the construction for b, the result follows directly. If this is not the case, this imposes a restriction on inputs and outputs for the construction of b: the difference (or linear mask) of every input and output subblock must be zero. In that case, the MILP tool proved that the four new F-functions that were added when going from b to $b+1$ ensure that there will be at least 30 active S-boxes. One such case is illustrated in Fig. 15.

B Efficient Implementation For $b = 1$

We recall the four Intel instructions to implement one round of AES: `AESENC` (Alg. 13), `AESENCLAST` (Alg. 14), `AESDEC` (Alg. 15), and `AESENCLAST` (Alg. 16). The `AESIMC` instruction corresponds to the `InvMixColumns` operation. Then for $b = 1$, Simpira can be implemented as in Alg. 17, and Simpira^{-1} as in Alg. 18.

C Optimizing the Number of F-functions

In Sect. 3, we based ourselves mainly on designs that were published in literature, instead of exhaustively searching for the optimal design that satisfies the design criteria. Here, we revisit this assumption. In particular, we will consider the case where $b = 4$.

An exhaustive search for all GFS with $b = 4$ shows that full bit diffusion requires at least 8 F-functions. One such construction is shown in Fig. 17. However, we found that all designs with 8 F-functions have a lower bound of at most 10 for the number of active S-boxes, and therefore do not satisfy our design criteria.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

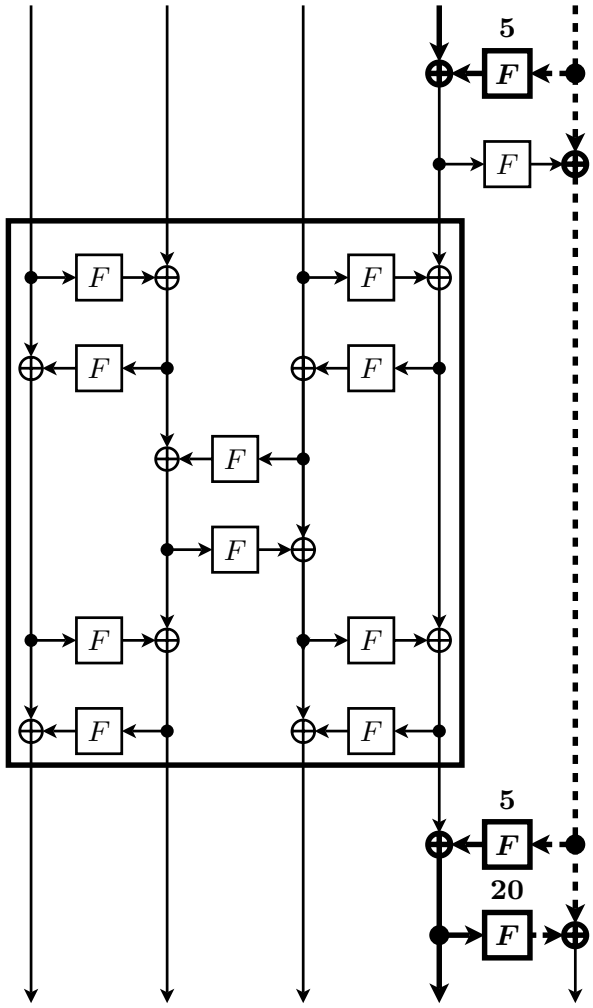


Fig. 15. A differential characteristic for (reduced-round) Simpira with $b = 5$ that has 30 active S-boxes. A thick full line indicates a difference in every byte, a thick dotted line refers to a difference in only one byte – it does not matter which one. A normal line indicates that no difference is present. When non-zero, the number of active S-boxes is shown above every F-function.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Algorithm 13 AESENC (= Alg. 1)

```
1: procedure AESENC(state, key)
2:     state ← SubBytes(state)
3:     state ← ShiftRows(state)
4:     state ← MixColumns(state)
5:     state ← state ⊕ key
6:     return state
7: end procedure
```

Algorithm 14 AESENCLAST

```
1: procedure AESENCLAST(state, key)
2:     state ← SubBytes(state)
3:     state ← ShiftRows(state)
4:
5:     state ← state ⊕ key
6:     return state
7: end procedure
```

Algorithm 15 AESDEC

```
1: procedure AESDEC(state, key)
2:     state ← InvSubBytes(state)
3:     state ← InvShiftRows(state)
4:     state ← InvMixColumns(state)
5:     state ← state ⊕ key
6:     return state
7: end procedure
```

Algorithm 16 AESDECLAST

```
1: procedure AESDECLAST(state, key)
2:     state ← InvSubBytes(state)
3:     state ← InvShiftRows(state)
4:
5:     state ← state ⊕ key
6:     return state
7: end procedure
```

Algorithm 17 Simpira ($b = 1$) (= Alg. 3)

```
1: procedure SIMPIRA(x_0)
2:     R ← 6
3:     for r = 1, ..., R − 1 do
4:         C ← SETR_EPI32(0x00 ⊕ r ⊕ R,
5:                        0x10 ⊕ r ⊕ R,
6:                        0x20 ⊕ r ⊕ R,
7:                        0x30 ⊕ r ⊕ R)
8:
9:         x_0 ← AESENC(x, C)
10:        x_0 ← AESENC(x, 0)
11:        c ← c + 1
12:    end for
13:    C ← SETR_EPI32(0x00 ⊕ R ⊕ R,
14:                   0x10 ⊕ R ⊕ R,
15:                   0x20 ⊕ R ⊕ R,
16:                   0x30 ⊕ R ⊕ R)
17:
18:    x_0 ← AESENC(x, C)
19:    x_0 ← AESENCLAST(x, 0)
20:    return x_0
21: end procedure
```

Algorithm 18 Simpira^{-1} ($b = 1$) (= Alg. 4)

```
1: procedure SIMPIRA(x_0)
2:     R ← 6
3:     for r = R, ..., 1 do
4:         C ← SETR_EPI32(0x00 ⊕ r ⊕ R,
5:                        0x10 ⊕ r ⊕ R,
6:                        0x20 ⊕ r ⊕ R,
7:                        0x30 ⊕ r ⊕ R)
8:         C ← AESIMC(C)
9:         x_0 ← AESDEC(x, C)
10:        x_0 ← AESDEC(x, 0)
11:        c ← c + 1
12:    end for
13:    C ← SETR_EPI32(0x00 ⊕ 1 ⊕ R,
14:                   0x10 ⊕ 1 ⊕ R,
15:                   0x20 ⊕ 1 ⊕ R,
16:                   0x30 ⊕ 1 ⊕ R)
17:    C ← AESIMC(C)
18:    x_0 ← AESDEC(x, C)
19:    x_0 ← AESDECLAST(x, 0)
20:    return x_0
21: end procedure
```

Fig. 16. In Alg. 13–18, we recall the AES-NI instructions of [44] and show how they can be used to efficiently implement Simpira and its inverse for $b = 1$.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

To reach both full bit diffusion and at least 25 active S-boxes, at least 9 F-functions are required. An example of such a construction is shown in Fig. 18. This construction has 35 active S-boxes, and would therefore lead to a slightly better construction that could replace the current choice for Simpira with $b = 4$.

However, the problem with this approach is all of these Feistels have a very random-looking structure. When there is no simple structure, the design becomes more difficult to cryptanalyze, and possibly also more difficult implement.

It therefore seems better to consider only GFS with either identical round functions ($b \in \{2, 3, 4, 6, 8\}$ of Simpira), or with the TwoF function that is used in Simpira for large b. With this restriction, the Simpira design for $b = 4$ with 10 F-functions is an optimal according to the design criteria.

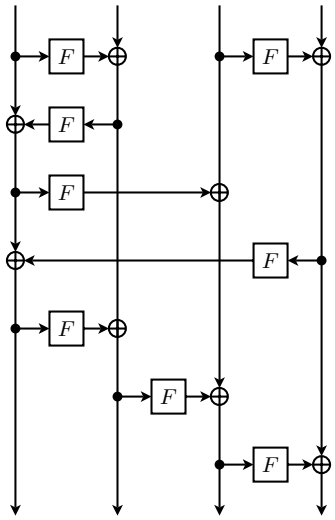


Fig. 17. A GFS with $b = 4$ that uses 8 F-functions to reach full bit diffusion. This design has at least 10 active S-boxes.

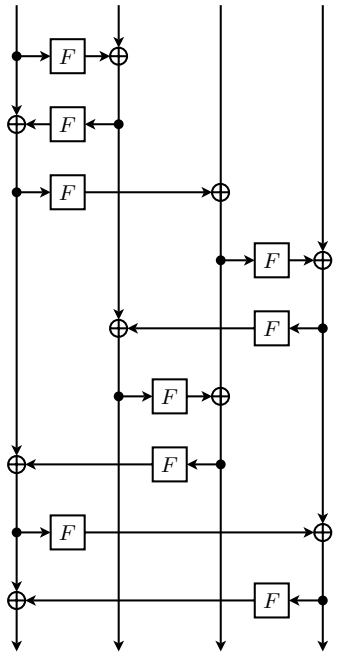


Fig. 18. A GFS with $b = 4$ that uses 9 F-functions to reach full bit diffusion. This design has at least 35 active S-boxes.

D Comparison with Other Constructions

For comparison, we now provide the throughput of SHA-256, SHA-512, and Rijndael256 (with a 256-bit block size), measured on the same platform, and using the same methodology. In the case of SHA-256 and SHA-512, we wrote

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

an optimized throughput-oriented implementation that uses the AVX2 architecture, available on the discussed platform. For SHA-256 and SHA-512, this implementation processes 4 and 8 independent (long) buffers respectively. For Rijndael256, we prepared optimized code that uses AES-NI (see details in [44]). We measured it in ECB mode, operating on 8 blocks in parallel, to get the highest throughput possible on this platform.

Under this setup, the throughput of SHA-256, SHA-512, and Rijndael256 is 2.35, 3.13, and 1.54 cycles per byte, respectively. Therefore, for $b = 2$, it is clearly much faster to use the Simpira permutation, which requires only 0.94 cycles per byte. This permutation is to be used inside an Even-Mansour construction (for encryption), or with a Davies-Meyer feedforward (for hashing); but these operations not change the throughput in a noticeable way.

For larger b, it is interesting to compare Simpira with two-pass constructions, for example for encryption. We cannot use AES as a building block in a typical two-pass construction, as it would be insecure beyond about 2^{64} input blocks, and we aim for security up to 2^{128} blocks. We may choose Simpira with $b = 2$ as a building block, as typical alternatives such as Rijndael256 are slower on our target platform.

In a double-pass mode of operation, Simpira with $b = 2$ requires at least $2 \cdot 30 \cdot b$ AESENC operations per $32b$ bytes, which is $30b$ AESENC operations per $16b$ bytes. When b is large, Simpira requires $24b - 36$ AESENC operations per $16b$ bytes, which is less than the previously mentioned double-pass mode of operation, even for very large b.

On a sidenote: Simpira with large b is faster than the two-pass construction, but cannot process the input blocks in parallel: to fill the pipeline, a sufficient number of independent messages is required. Therefore, which of these two Simpira-based constructions is better, depends on the application.

Bhaumik, Ritam; Datta, Nilanjan; Dutta, Avijit; Mouha, Nicky; Nandi, Mrudil.
"Simpira v2: A Family of Efficient Permutations Using the AES Found Function."
Paper presented at the 22nd Annual International Conference on the Theory and Application of Cryptology and Information Security,
ASIACRYPT 2016, Hanoi, Viet Nam. December 4, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Identifying Evidence for Implementing a Cloud Forensic Analysis Framework

Changwei Liu[#], Anoop Singhal[*], Duminda Wijesekera [#,*]

cliu6@gmu.edu, anoop.singhal@nist.gov, dwijesek@gmu.edu

[#] Department of Computer Science, George Mason University, Fairfax VA 22030 USA

[*] National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg MD 20899 USA

Abstract: Cloud computing provides several benefits to organizations such as increased flexibility, scalability and reduced cost. However, it provides several challenges for digital forensics and criminal investigation. Some of these challenges are the dependence of forensically valuable data on the deployment model, multiple virtual machines running on a single physical machine and multiple tenancies of clients. In this paper, we show what evidence from the cloud would be useful to construct the attack scenario by using a Prolog logic based forensic analysis tool. We propose to implement and design a forensic enabled cloud, which includes installing forensic tools in the cloud environment and logging all the activities from both the application layer and lower layers. Such an implementation can provide evidence for a Prolog based forensic tool, which can automate correlating the evidence from both the clients and the cloud service provider to construct attack steps and therefore re-create the attack scenarios on the cloud.

Keyword: Digital forensic analysis, cloud forensics, attack scenario, OpenStack

Liu, Changwei; Singhal, Anoop; Wijesekera, Duminda.
"Identifying Evidence for Implementing a Cloud Forensic Analysis Framework."
Paper presented at Thirteenth IFIP WG 11.3 International Conference on Digital Forensics, Orlando, FL. January 30, 2017 - February 1, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

1. Introduction

Digital forensics is the application of science to identify, collect, examine, and analyze data while preserving information integrity and maintaining a strict chain of custody for the data during post incident examinations [1]. Being a component of digital forensics, network forensics analyzes network traffic in order to gather information from intrusion detection systems or logs to constitute legal evidence [2]. Considered as an emerging branch of network forensics, cloud forensics involves post-incident analysis of systems with distributed processing, multi-tenancy, virtualization and mobility of computations, which poses more challenges in identifying and preserving digital evidence, including [3]:

1. Dependence of forensically valuable data on the deployment model and methods. For example, customers of software as a service (SaaS) may have little or no control of the physical locations of their data.
2. Large volume in content and proprietary formats of data logs.
3. The diversity and the number of simultaneously operating virtual machines instances of a single physical machine isolated using virtualization. This takes extra efforts in segregating resources without breaching user confidentiality. In addition, weak registries in clouds make it easy for attackers to hide their traces.
4. Instances of servers running on virtual machines in the cloud monitored by hypervisors lack of warnings, procedures and tools for forensic investigation.

Although much research has progressed in digital forensics, the methods used in traditional digital forensics are inadequate for forensic investigation in clouds that have

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

been designed without much efforts dedicated for evidence retention and integrity. Recently, National Institute of Standards and Technology (NIST) and other researchers have published papers in cloud governance, security and risk assessment [4], and proposed implementing forensic-enabled clouds. For example, Dykstra et al. proposed implementing cloud to collect forensic data from operating system level underneath the virtual machines [2]. Zawod et al. provided complete, trustworthy, and forensic-enabled cloud architecture to collect logs for forensic analysis [3]. However, these implementations only focus on evidence acquaintance on Infrastructure-as-a-Service (IaaS) cloud deployment model. None of the work has discussed implementing forensic-enabled clouds that cover all deployment models. In this paper, we show what evidence can be used to construct corresponding attack scenarios in the cloud, and discuss how we may implement and automate the forensic analysis in the cloud using some example attacks with the objective of creating a deployment model.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 shows our experimental attacks in the cloud, and how we identify the evidence from the cloud to construct attack scenario by using a Prolog based tool. Section 4 shows how to use system call sequence to construct attack steps when other evidence is unavailable. We conclude the paper by discussing how we may implement and automate the forensic analysis in the cloud as our future work in Section 5.

2. Background and Related Work

We present the background and research related to digital and cloud forensics in this section.

2.1 Digital Forensics

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Digital forensics uses scientifically accepted methods to collect, validate and preserve digital evidence derived from digital sources for the purpose of reconstruction of events found to be criminal or helping to anticipate unauthorized actions shown to be disruptive to planned operations [5]. Digital forensic investigators seek attack evidence from computers and networks. Typically, imaging tools are used to extract a computer's physical memory or disk sectors to a file, and then the investigators feed the file into data analysis tools to perform live or dead analysis. For network evidence, forensic investigators analyze network traffic and gather information from intrusion detection systems or logs to constitute legal evidence.

2.2 Cloud Forensics

NIST defined cloud model [6] uses three service deployment models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). SaaS allows consumers to use the provider's applications running on a cloud infrastructure. PaaS allows consumers to deploy on the cloud consumer-created or acquired applications using programming languages, libraries, services and tools supported by the provider. IaaS provides consumers with the capability of provisioning processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software including operating systems and applications.

According to Ruan et al., cloud forensics is a subset of network forensics [3] that follows the main phases of network forensics with techniques tailored to cloud computing environments. For example, data acquisition is different in SaaS and IaaS, because the investigator will have to solely depend on cloud service provider in SaaS. With IaaS, the investigator can acquire the virtual machine image from customers.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2.3 Related Work

According to many researchers, data acquisition is a main issue in cloud forensics. Many methods have been proposed to collect evidence from clouds, which include remote data acquisition, management plane, live forensics and snapshot analysis [7]. Dykstra et al. successfully retrieved volatile and nonvolatile data from Amazon EC2 cloud's active user instance platform using forensic tools such as Guidance EnCase and Access Data FTK [8]. However, those tools do not validate data integrity. Researchers recommended and developed some toolkits to collect related logs from cloud infrastructure while assuring their integrity. Assuming the cloud provider is trustworthy, Dykstra et al. developed the FROST toolkit that can be integrated to OpenStack [9] to collect logs from the operating system level supporting the virtual machines [10]. Zawod et al. designed complete, trustworthy, and forensic-enabled cloud architecture for log collection to address this trust issue [11]. Hay et al. proposed live digital forensics analysis on clouds using virtual introspection, a process by which the state of a virtual machine (VM) is observed from either the hypervisor (VMM) or from some other virtual machine and presented a suite of virtual introspection tools developed for Xen (VIX tools) [12]. However, live forensic tools have not been incorporated and provided as a commercial service by the cloud service providers. Snapshot technologies enable customers to freeze a specific state of VM [13]. The snapshot images can be restored by loading them to a target VM for analysis, gaining information on the running state of a virtual machine that is supported by hypervisor vendors, including Xen, VMWare, ESX, Hyper-V, and cloud providers that support snapshot features.

In addition, many tools like Encase, the Sleuth Kit, SNORT, WireShark can collect

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

digital evidence from computers and networks. In order to reduce the investigators' time and effort in constructing attack steps, researchers proposed using rules to automate correlating evidence by finding the causality between items of evidence [14, 15]. Liu et al. integrated the tool with two databases, including a vulnerability database and an anti-forensic database, to ascertain the admissibility of evidence and explain missing evidence due to attackers' using anti-forensics [15]. These rule based forensic analysis frameworks have been proposed for network forensics, but have not been tested in a cloud environment.

3. Using Alerts and Logs to Construct Attack Scenario

In this section, we describe experimental attacks we launched on OpenStack [9] to identity evidence that can be used for cloud forensic analysis.

3.1 Experimental Environment Setup

OpenStack is a collection of python-based software projects that manage access to pooled storage, computing and network resources that reside in one or multiple machines of a cloud. This collection has six core projects: Neutron (Networking), Nova (Compute), Glance (Image Management), Swift (Object Storage), Cinder (Block Storage) and Keystone (Authorization and Authentication) [9]. OpenStack can be used to deploy three service models--SaaS, PaaS and IaaS, but is mostly deployed as IaaS.

"DevStack" is a series of extensible scripts that can invoke an OpenStack environment using the latest versions of the software. We deployed OpenStack "Juno" version as IaaS cloud on a single computer (with IP address 172.16.168.100) by running "Devstack" on an Ubuntu 14.04 Desktop. Authenticated users can access OpenStack services using the IP address 172.16.168.100 on their browser to access the control dashboard Horizon as shown in Figure 1.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

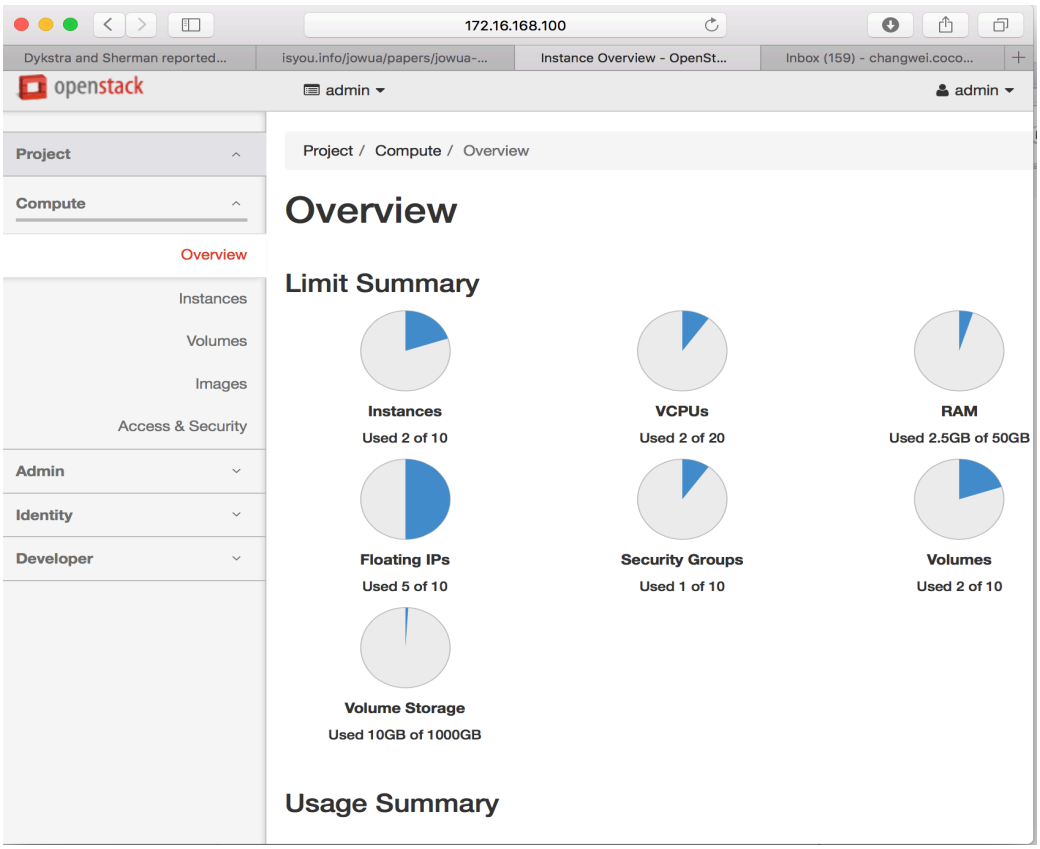
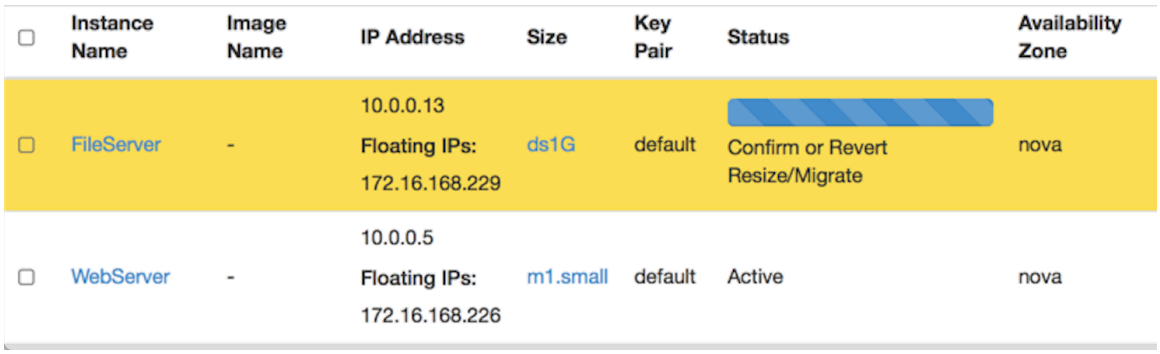


Figure 1. OpenStack web user interface--Horizon

We deployed two virtual machines (also called running instances), a webserver (named "WebServer" associated with IP address 172.16.168.226) and a fileserver (named "FileServer" associated with IP address 172.16.168.229) under the authenticated user "Admin" in our OpenStack cloud. In the "WebServer", we deployed an Apache webserver and a MySQL database, allowing users to query their data using the webserver. Authenticated users can access the "FileServer" by remotely using "ssh". In order to launch an attack, we also installed Kali (the penetration testing and ethical hacking Linux distribution tool [16]) in the same network (with IP address 172.16.168.173).

3.2 Example Attacks

We launched three attacks, a SQL injection attack, a DDoS attack, and a DoS attack towards the two VMs in our IaaS cloud.



	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone
☐	FileServer	-	10.0.0.13 Floating IPs: 172.16.168.229	ds1G	default	Confirm or Revert Resize/Migrate	nova
☐	WebServer	-	10.0.0.5 Floating IPs: 172.16.168.226	m1.small	default	Active	nova

Figure 2. Resizing “FileServer”

Our SQL injection attack exploits un-sanitized user inputs (CWE89) in the “WebServer”. Our DDoS attack known as "TCP connection flood" used “nping” in Kali to flood the “FileServer” in order to prevent legitimate requests. While SQL injection and DDoS attacks can happen to any network including a cloud that has corresponding vulnerability, only IaaS privileged users can resize and delete a VM by launching DoS attacks that exploit the vulnerability “CVE-2015-3241”. According to NIST’s NVD, the vulnerability “CVE-2015-3241” that is in OpenStack Compute (Nova) versions 2015.1 through 2015.1.1, 2014.2.3 allows authenticated users to cause denial of services by resizing and then deleting an instance (VM). The process of resizing and deleting an instance in this way is also called instance migration. With “CVE-2015-3241”, the migration process does not terminate when an instance is deleted, so an authenticated user could bypass user quota enforcement to deplete all available disk space by repeatedly performing instance migration. Figure 2 shows the process of our resizing the file server from “ds512M” to “ds1G”, where we can see the instances’ availability zone is “nova”. We continued to resize and delete instances until Nova was so depleted that it could not accept any new instance.

3.3 Identifying Evidence to Reconstruct Attack Scenarios

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

In order to obtain evidence for forensic analysis, we configured the webserver and the SQL database in “WebServer” to log access and query history. We also installed Snort in “WebServer” and “FileServer” VMs and deployed WireShark in the host Ubuntu OS to monitor the network traffic. Snort was able to capture the SQL injection attack and generated alerts with appropriate rules. Also, WireShark was able to capture packets that formed the DDoS attack. Figure 3 lists some SNORT alerts and MySQL query log of the SQL injection attack, which shows the attack was done by using “ or ‘1’=‘1’ ” to bypass the SQL query condition check. The snapshot of packets captured by WireShark is listed in Figure 4, where we can see Kali Linux at 172.16.168.173 sent out numerous SYN packets to “FileServer” at 172.16.168.229, and the “FileServer” sent numerous SYN-ACK packets back to Kali Linux.

```
 [**] SQL Injection Attempt --1=1 [**]
08/16-14:37:27.818279 172.16.168.173:1715 -> 172.16.168.226:80
TCP TTL:128 TOS:0x0 ID:380 IpLen:20 DgmLen:48 DF
******S* Seq: 0xDEDBEABF  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK


160813 14:37:29 40 Connect
...
40 QuerySET GLOBAL general_log = 'ON' 40 Queryselect * from profiles where
name='Alice' AND password='alice' or '1'='1'
Gen_log 2: 130813 14:39:56
…
```

Figure 3. The SNORT alert and the MySQL database log

No.	Time	Source	Destination	Protocol	Length	Info
217	10.405625326	172.16.168.173	172.16.168.229	TCP	74	34818 → 80 [SYN] Seq=0
218	10.405682554	172.16.168.173	172.16.168.229	TCP	74	44208 → 80 [SYN] Seq=0
219	10.405746104	172.16.168.173	172.16.168.229	TCP	74	38032 → 80 [SYN] Seq=0
220	10.408041819	172.16.168.173	172.16.168.229	TCP	74	34348 → 80 [SYN] Seq=0
221	10.408111539	172.16.168.173	172.16.168.229	TCP	74	38769 → 80 [SYN] Seq=0
222	10.408205849	172.16.168.173	172.16.168.229	TCP	74	36846 → 80 [SYN] Seq=0
223	10.408275950	172.16.168.173	172.16.168.229	TCP	74	35307 → 80 [SYN] Seq=0
224	10.408329211	172.16.168.229	172.16.168.173	TCP	60	80 → 41930 [RST, ACK]
225	10.408355690	172.16.168.229	172.16.168.173	TCP	60	80 → 44471 [RST, ACK]
226	10.408388686	172.16.168.173	172.16.168.229	TCP	74	35276 → 80 [SYN] Seq=0
227	10.408430802	172.16.168.229	172.16.168.173	TCP	60	80 → 45714 [RST, ACK]
228	10.408465024	172.16.168.229	172.16.168.173	TCP	60	80 → 35431 [RST, ACK]

Figure 4: A Snippet of packets caught by WireShark

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

```
/*The initial attack status and final attack status*/
attackerLocated(internet).
attackGoal(serviceDown(fileServer,user)).
attackGoal(execCode(database,user)).

/* The network topology and computer configuration*/
/* "_" means any port */
hacl(internet, webServer, tcp, 80).
hacl(internet, fileServer, tcp, _).
directAccess(webServer,database,modify,user).

/* The evidence found in webServer */
vulExists(webServer, 'SQLInjection', httpd).
vulProperty('SQLInjection', remoteExploit, privEscalation).
networkServiceInfo(webServer , httpd, tcp , 80 , user).

/* The evidence captured by WireShark*/
vulExists(fileServer,'DDoS', httpd).
vulProperty('DDoS', remoteExploit, privEscalation).
networkServiceInfo(fileServer, httpd, tcp, _, user).
```

Figure 5. Prolog predicates for SQL injection and DDoS attack evidence

We used our Prolog based tool presented in [15] to automate the process of correlating items of evidence to generate attack scenarios. To do so, we converted the above evidence and the cloud configuration to corresponding Prolog predicates as the input file in Figure 5. Our Prolog based tool uses rules to correlate these items of evidence represented by Prolog predicates to construct attack paths. The constructed attack paths are shown in Figure 6, and the notation of all nodes is listed in Table 1. In this graph model, an attack status obtained from the attacked system is represented by a diamond. All computer configuration, network topology and software vulnerabilities used to launch an attack are represented by boxes and a rule used to correlate attack status before and after an attack step is represented by an ellipse [15]. The software vulnerability exploited to launch an attack is obtained by forensic investigators' judgment on the evidence collected from the attacked system. Two attack paths are shown in Figure 6. The left path

(8⟶6⟶5⟶4⟶3⟶2⟶1) represents the SQL injection attack that used the web server vulnerability to maliciously obtain the information from the MySQL database. The right path (8⟶15⟶14⟶13⟶12) represents the DDoS attack to bring down the "FileServer".

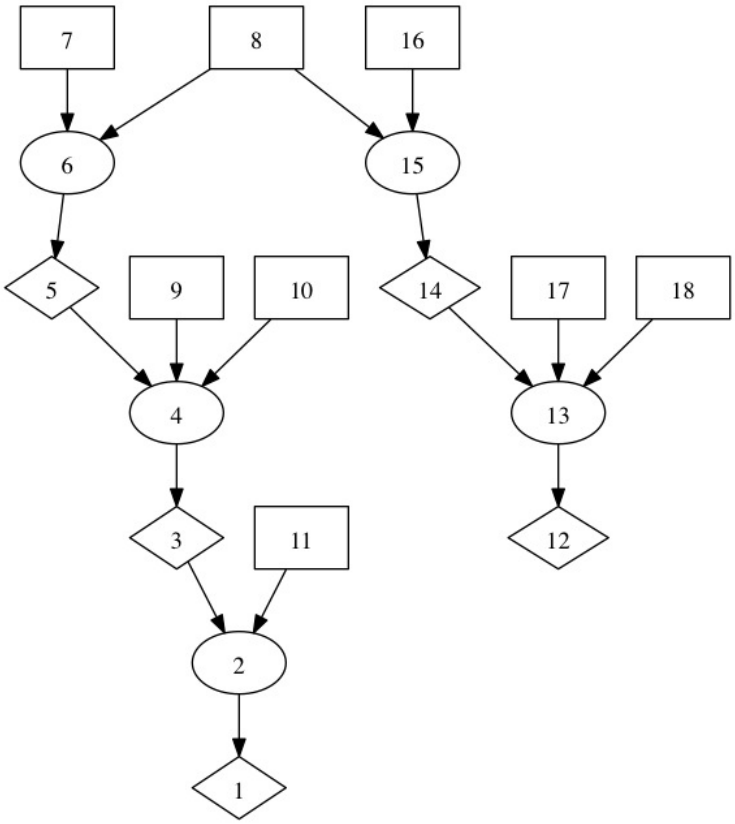


Figure 6. The attack path constructed for SQL injection and DDoS attacks

SNORT and WireShark failed in capturing our DoS attack on the "FileServer" that exploited the "CVE-2015-3241" vulnerability on OpenStack Nova service. Because OpenStack service application programming interface (API) logs provide information about users' operations on the running instances, we used the OpenStack service API logs as evidence. Figure 7 lists a snippet of Nova API logs that are related to our instance migration of the DoS attack, where the commands in bold font show the instance

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

"bd1dac18-1ce2-44b5-93ee-967fec640ff3" representing the "FileServer" VM (as shown in Table 2, which is obtained by running "nova list" in the Ubuntu host system.) has been resized using commands "mv" (move) and "mkdir" (create new directory) operated by the user "admin". We aggregated the related Nova API calls as evidence to form the input file with the corresponding attack status and the system configuration (Figure 8). By running our Prolog based tool [15] on the input file, we obtained the attack scenario as shown in Figure 9 with the notation of all nodes in Table 3. The figure shows the attack path that used the control dashboard "Horizon" exploiting the "CVE-2015-3241" vulnerability.

Table 1. Notation of all nodes in Figure 6

Node Number	Notation of the Node
1	execCode(database,user)
2	THROUGH 7 (Attack by compromised computer)
3	execCode(webServer,user)
4	THROUGH 3 (remote exploit of a server program)
5	netAccess(webServer,tcp,80)
6	THROUGH 9 (direct network access)
7	hacl(internet,webServer,tcp,80)
8	attackerLocated(internet)
9	networkServiceInfo(webServer,httpd,tcp,80,user)
10	vulExists(webServer,'SQLInjection',httpd,remoteExploit,privEscalation)
11	directAccess(webServer,database,modify,user)
12	execCode(fileServer,user)
13	THROUGH 3 (remote exploit of a server program)
14	netAccess(fileServer,tcp,_)
15	THROUGH 9 (direct network access)
16	hacl(internet,fileServer,tcp,_)
17	networkServiceInfo(fileServer,httpd,tcp,_,user)
18	vulExists(fileServer,'DDoS',httpd,remoteExploit,privEscalation)

2016-09-18 07:52:00.237 DEBUG oslo_concurrency.processutils [req-f79c7911-04ed-4a0c-adbe-0ae0a487c0f7 admin admin] Running cmd (subprocess): **mv /opt/stack/data/nova/instances/bd1dac18-1c e2-44b5-93ee-967fec640ff3 /opt/stack/data/nova/instances/bd1dac18-1ce2-44b5-93ee-967fec640ff3_resize from (pid=41737) execute /usr/local/lib/python2.7/dist-packages/oslo_concurrency/processutils.py:344**

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2016-09-18 07:52:00.253 DEBUG oslo_concurrency.processutils [req-f79c7911-04ed-4a0c-adbe-0ae0a487c0f7 admin admin] CMD **"mv /opt/stack/data/nova/instances/bd1dac18-1ce2-44b5-93ee-967fec640ff3 /opt/stack/data/nova/instances/bd1dac18-1ce2-44b5-93ee-967fec640ff3_resize"** returned: 0 in 0.016s from (pid=41737) execute /usr/local/lib/python2.7/dist-packages/oslo_concurrency/processutils.py:374

2016-09-18 07:52:00.254 DEBUG oslo_concurrency.processutils [req-f79c7911-04ed-4a0c-adbe-0ae0a487c0f7 admin admin] Running cmd (subprocess): **mkdir –p /opt/stack/data/nova/instances/bd1dac18-1ce2-44b5-93ee-967fec640ff3** from (pid=41737) execute /usr/local/lib/python2.7/dist-packages/oslo_concurrency/processutils.py:344

2016-09-18 07:52:00.271 DEBUG oslo_concurrency.processutils [req-f79c7911-04ed-4a0c-adbe-0ae0a487c0f7 admin admin] CMD **"mkdir –p /opt/stack/data/nova/instances/bd1dac18-1ce2-44b5-93ee-967fec640ff3"** returned: 0 in 0.017s from (pid=41737) execute /usr/local/lib/python2.7/dist-packages/oslo_concurrency/processutils.py:374

Figure 7. Nova API Call Logs

Table 2. The VM instance IDs, names and IPs

ID	Name	…	Networks
bd1dac18-1ce2-44b5-93ee-967fec640ff3	FileServer		private=10.0.0.13, 172.16.168.229
c01d5e66-c20d-4544-867b-d3e2b70bfc60	WebServer		private=10.0.0.5, 172.16.168.226

/* the initial and final attack status*/
attackerLocated(controlDashboard).
attackGoal(execCode(nova,admin)).

/* the fileserver VM could be reached from control dashboard*/
hacl(controlDashboard, fileServer, http, _).

/* the evidence of attack using 'CVE-2015-3241' that uses RESTful service*/
vulExists(nova,'CVE-2015-3241', 'REST').
vulProperty('CVE-2015-3241', remoteExploit, privEscalation).
networkServiceInfo(nova, 'REST', http, _, admin).

Figure 8. The input file for attack using "CVE-2015-3241"

Figure 6 and Figure 9 cannot be grouped together, because attackers were in different locations. In addition, in Figure 9, the attack happened in the cloud compute service instead of a VM, although the attacker launched the attack from a VM. This is because all VMs share the same compute service in our cloud.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

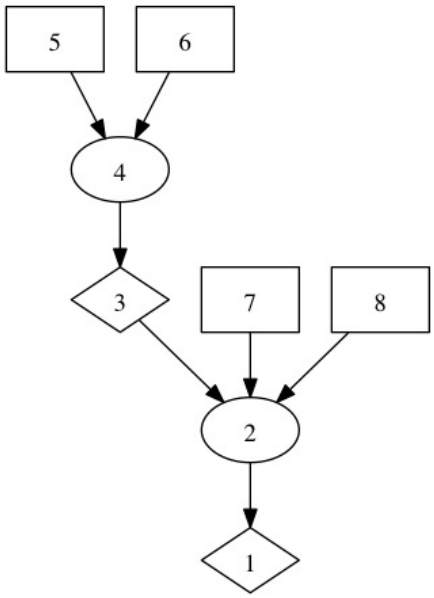


Figure 9. The attack path constructed for the DoS attack

Table 3. The notation of all nodes in Figure 9

Node Number	Notation of the Node
1	execCode(nova,admin)
2	THROUGH 3 (remote exploit of a server program)
3	netAccess(nova,http,_)
4	THROUGH 9 (direct network access)
5	hacl(controlDashboard,nova,http,_)
6	attackerLocated(controlDashboard)
7	networkServiceInfo(nova,'REST',http,_,admin)
8	vulExists(nova,'CVE-2015-3241','REST',remoteExploit,privEscalation)

4. Using System Call Invocations for Evidence Analysis

Because system calls allow user level processes to request kernel level services including access to storage operations, memory or network access, and process management, system calls sequence is often used for intrusion detection and forensics [17]. When evidence or expert knowledge is unavailable to recognize the interaction between user level processes to kernel level services as a known attack, forensic investigators analyze the system calls to ascertain program behavior. According to [18], it is rare or

unlikely to have an attack path, in which every attack step is a zero-day attack. As such, we use system calls to construct the missing attack steps only when other evidence is not available.

There are several popular mechanisms to trace the system calls in a cloud based VM: (1) use "ptrace" command to set up system call interception and modification by modifying a software application, (2) use "strace" command to log system calls and signals, (3) use auditing facilities within the kernel, (4) modify the system call table and write system call wrappers to log the corresponding system calls, (5) intercept the system call within the hypervisor [19]. Because OpenStack supports different hypervisors, including Xen, QEMU, KVM, LXC, Hyper-V and UML, there is no a generic solution to intercept the system call within the hypervisor. Thus, we use methods 2 and 4 to log relevant system calls.

```
Sep 25 00:15:49 FileServer sshd[829]: Server listening on 0.0.0.0 port 22.
Sep 25 00:15:49 FileServer sshd[829]: Server listening on :: port 22.
Sep 25 00:28:15 FileServer sshd[1162]: Accepted password for coco from 172.16.168.173 port 44842 ssh2
Sep 25 00:28:16 FileServer sshd[1162]: pam_unix(sshd:session): session opened for user coco by (uid=0)
```

Figure 10. The authentication log for sshd

Now we show how to use system call sequences to construct an attack step by using an attack example. In this experimental attack launched from our Kali Linux, we, as the attacker, used ssh to log into "FileServer" by using stolen credentials from a legitimate user named "coco". In order to simulate the stealthy attack without triggering IDS alerts, we assumed that the attacker could use social engineering attacks, such as shoulder surfing, to obtain the legitimate user's (username, password) pair to log into the "FileServer" using ssh. The corresponding sshd log from "/var/log/auth.log" in "FileServer" is listed in Figure

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

10, where user "coco" was listed to log in "FileServer" from "172.16.168.173" that actually belonged to the attacker, which indicates that the attacker stole user coco's credentials.

A process typically comprises of many system calls, of which only some generally are important to ascertain a process' behavior (we use the ones presented in [18]. These system calls are listed in the second column of Table 4). Figure 11 is a snippet of important system calls captured from the attack of using coco's stolen credentials to modify a file in "FileServer" (due to space limitations, we list a part of captured system calls). By analyzing these system calls, we notice that the "write/read" system calls (in bold font) indicate that the attacker used "vi test.txt"("vi" is a text editor) command to modify "test.txt" file. In the "write/read" system call, the first argument is the file descriptor where the process reads or writes, the second argument represents the content in the buffer, the third argument represents how many bytes the system call will write/read, and "=1/<any number greater than 1>" indicates that the system call executed successfully.

Table 4: Important System Calls

Tasks	System Calls
Process modifies file	write, pwrite64,rename, mkdir, linkat, link, symlinkat, symlink, fchmodat, fchmod, chmod, fchownat, mount
Process uses but does not modify file	stat64,lstat6e,fsat64, open, read, pread64, execve, mmap2, mprotect, linkat, link, symlinkat, symlink
Process uses and modifies file	open, rename, mount, mmap2, mprotect
Process creation or termination	vfork, fork, kill
Process creation	Clone

write(9, "v", 1) = 1
read(11, "v", 16384) = 1
write(3, "\0\0\0\20\331\255\275\264c\2173)z2j\32\255
n\2007d\366m\21\316\2648\240\207\31\211"..., 36) = 36
read(3,
"\0\0\0\20\240\253\341\227\321xU\305\347\226\246\361\316\242S=\30\341QT\231\n\343\314\34
3\307\f\361"..., 16384) = 36
write(9, "i", 1) = 1

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

```
read(11, "i", 16384)                    = 1
write(3,
"\0\0\0\20\177\352\313\332\373yjM\3416l\230\215\10\220p\252g\375\365\1\f\335\361\r\273\374\
357"..., 36) = 36
read(3,
"\0\0\0\20\27\334?\201x\300\16\356\346,\0379\32\220{\372)\366\4\v\1=\347\263\311\250k\353"...
, 16384) = 36
write(9, " ", 1)                        = 1
read(11, " ", 16384)                    = 1
write(3,
"\0\0\0\20`i\321\344\220\313\322\254S\252o\201\225;6v\243\205\10gs^\253\237\325\375\332v"...
, 36) = 36
read(3, "\0\0\0\20\5\27k;\254\301\24\n\\ZN\267\260\336\323`\323\32\345\2b\226-\271|[B\21"...,
16384) = 36
write(9, "t", 1)                        = 1
read(11, "t", 16384)                    = 1
read(3,
"\0\0\0\20\325\261\7\254\211(\201\331\272\344[\355\200\\u4\357G\347\232\276:\201\376\342\20
2\201."..., 16384) = 36
write(3,
"\0\0\0\20\320\254#\312\211_\3022\n\227u\16I\372\202\347\37\252T\257\220\210E\343\222\342\
24S"..., 36) = 36
write(9, "e", 1)                        = 1
read(11, "e", 16384)                    = 1
write(3, "\0\0\0\20\334n}4\375Q\212o\353\375\262\342\316\334w-
F\213\303\277t\312\245\16\266\255B|"..., 36) = 36
read(3, "\0\0\0\20\274\376\7J\214L\314OL\1c\22\364-gvJ%\21\344J<,h\363\261\36\10"..., 16384)
= 36
write(9, "\t", 1)                       = 1
read(11, "st.txt ", 16384)              = 7
…
```

Figure 11. Traces of “Read” and “Write” system calls

```
//The initial attack status
attackerLocated(internet).
// the attacker was able to log into “FileServer” by using stolen credentials
attackGoal(logInService(fileserver, tcp,22)
attackGoal(princinpalCompromised(user))
//InCompetent user
InCompetent(user).

//The attack status obtained from analyzing system call sequence
attackGoal(canAccessFile(fileServer,user,modify,_)).
//The user could login fileserver by using ssh protocol
networkServiceInfo(fileServer , sshd, tcp, 22, _).
//the user who has the account on “FileServer” has the privilege to modify a file
localFileProtection(fileServer,user,modify,_).
```

Figure 12. Input file for the attack of modifying a file with stolen credentials

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

We converted the program behavior learned from the system call sequence in Figure 11, that is the attacker's using a text editor to convert "test.txt" file, to Prolog Predicate "canAccessFile(fileServer,user,modify,_)" (This predicate means that the attacker as the user can modify the file located at "_" representing the home directory of the user). With the evidence obtained from the log in Figure 10 that the attacker with stolen credentials (represented by predicates "attackGoal(princinpalCompromised(user))", "InCompetent(user)" and "attackerLocated(internet)") logged into the "FileServer" by using ssh (represented by Predicate "attackGoal(logInService(fileserver, tcp,22)"), and the fact user "coco", who has an account on "FileServer", has the privilege to modify a file (the corresponding predicate is "localFileProtection(fileServer,user,modify,_)"), we formed the input file as Figure 12 to use our Prolog based tool. The constructed attack path is shown in Figure 13, and the notation of all nodes is in Table 5. In Figure 13, the attack step (3, 4, 7) ⟶ 2⟶1 has two pre-conditions represented by Node 4 and Node 7. Node 4 is obtained from the fact that the "FileServer" can be accessed by using ssh with protocol tcp from port 22. Node 7 is obtained from ssh authentication log in Figure 10 that indicates the user's credentials have been stolen by the attacker. Without the evidence obtained from the system call sequence (Node 1), the attack step (3, 4, 7) ⟶ 2⟶1 would not have been established.

Notice the two rule nodes (Node 5 and Node 2) in Figure 13 do not have any rule description because of the obvious correlation between Node 6 and Node 4 (if the network provides the service of using ssh to log into a fileserver by using tcp at port 22, the user including the attacker could log into the fileserver with stolen credentials), nodes (3,4,7) and Node 1 (if a user is allowed to have the privilege of modifying a file in fileserver, the

attacker with the stolen credentials from the user could access the file and modify it).

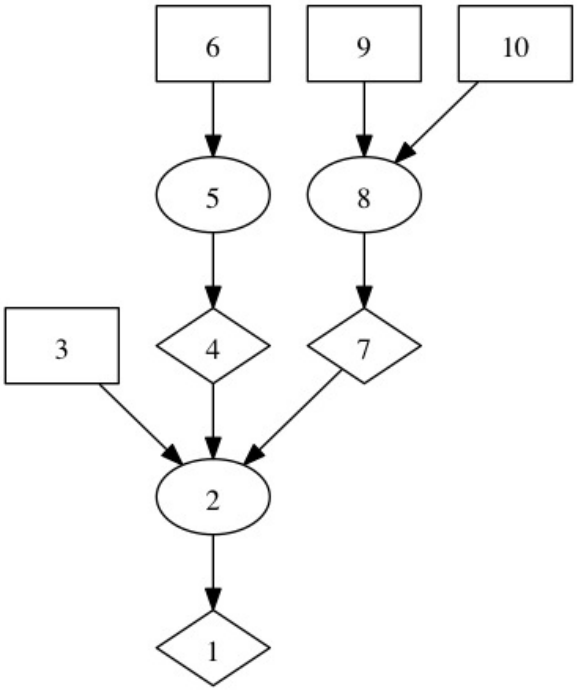


Figure 13. The attack step constructed by using evidence obtained from system calls

Table 5. The notation of all nodes in Figure 13

Node Number	Notation of Node
1	canAccessFile(fileserver,user, modify,_)
2	THROUGH 23()
3	localFileProtection(fileserver,user,modify,_)
4	logInService(fileserver,tcp,22)
5	THROUGH 18 ()
6	networkServiceInfo(fileserver,sshd,tcp,22,user)
7	princinpalCompromised(user)
8	THROUGH 16(password sniffing)
9	inCompetent(user)
10	attackerLocated(internet)

5. Conclusion and Future work

The use of cloud computing can increase the flexibility and efficiency of organizations or enterprises. However, clouds present significant challenges to forensics,

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

including customers' lack of control of the physical locations of their data, the large volume of data logs and the prevalence of proprietary formats. To solve the above problems, we plan to implement a forensic-enabled cloud by exploring what evidence could be useful for cloud forensic analysis.

Our example attacks show evidence from three resources could help investigators to construct attack scenarios, which include (1) evidence from IDS and application software logging, (2) cloud service API calls, and (3) system calls from VMs. To acquire the evidence from the three resources, the forensic-enabled cloud should have three extensions, which can (1) retrieve IDS and software service logging; (2) store and secure OpenStack service API call logs, firewall logs and snapshots for running instances; (3) obtain system calls when the evidence from (1) and (2) is missing. Our future plan addresses implementing forensic-enabled cloud with the above extensions and resolving the corresponding problems including assuring the data integrity, reducing the large volume and formalizing the proprietary of forensic data stored in the cloud.

DISCLAIMER

This paper is not subject to copyright in the United States. Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

[1] Kent K, Chevalier S, Grance T and Dang H. "Guide to integrating forensic techniques into incident response". 2006. p. 800e86. NIST Special Publication.

[2] Gary Palmer. "A Road Map for Digital Forensic Research". Report from DFRWS 2001,

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

First Digital Forensic Research Workshop, Utica, New York, August 7 – 8, 2001, Page(s) 27–30.

[3] Ruan, Keyun, Joe Carthy, Tahar Kechadi, and Mark Crosbie. "Cloud forensics." In IFIP International Conference on Digital Forensics, pp. 35-46. Springer Berlin Heidelberg, 201.

[4] M. Hogan, F. Liu, A. Sokol,J. Tong. "NIST cloud computing standards roadmap." NIST Special Publication 35 (2011).

[5] A. Jaquith, "Security Metrics: Replacing Fear, Uncertainty, and Doubt", Addison Wesley, Mar 26, 2007.

[6] P. Mell and T. Grance. "NIST definition of cloud computing". National Institute of Standards and Technology. October 7, 2009.

[7] Pichan, Ameer, Mihai Lazarescu, and Sie Teng Soh. "Cloud forensics: technical challenges, solutions and comparative analysis." Digital Investigation 13 (2015): 38-57.

[8] J. Dykstra and A.T. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques," in Proc. of the 12th Annual Digital Forensics Research Conference (DFRWS'12), Washington, DC, USA, Digital Investigation, vol. 9, August 2012, pp. 90–98.

[9] OpenStack Open Source Cloud Computing Software. Retrieved from https://www.openstack.org.

[10] J. Dykstra and A. Sherman, "Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform," Digital Investigation, vol. 10, no. Supplement, p. S87–S95, 2013.

[11] S. Zawoad and R. Hasan, "FECloud: A Trustworthy Forensics-Enabled Cloud Architecture," Proc. 11th Ann. Int'l Fed. Info. Processing WG 11.9 Int'l Conf. Digital Forensics, 2015, pp. 271–285.

[12] Hay, Brian, and Kara Nance. "Forensics examination of volatile system data using virtual introspection." ACM SIGOPS Operating Systems Review 42.3 (2008): 74-82.

[13] D. Birk, C. Wegener. "Technical issues of forensic investigations in cloud computing environments". In 6th International workshop on systematic approaches to digital forensic engineering–IEEE/SADFE 2011, Oakland, CA, USA; 2011, p. 1–10.

[14]] W. Wang, E.D. Thomas, "A graph based approach toward network forensics analysis", ACM Transactions on Information and Systems Security 12 (1) 2008.

[15] C. Liu, A. Singhal, D. Wijesekera. "A Logic Based Network Forensics Model for Evidence Analysis". IFIP Int. Conf. Digital Forensics 2015.

[16] Kali Linux--Penetration Testing and Ethical Hacking Linux Distribution. Retrieved from https://www.kali.org.

[17] Hofmeyr, Steven A., Stephanie Forrest, and Anil Somayaji. "Intrusion detection using sequences of system calls." Journal of computer security 6, no. 3 (1998): 151-180.

[18] X. Sun, J. Dai, A. Singhal, P. Liu and J. Yen, “Towards Probabilistic Identification of Zero-day Attack Paths”, Accepted for IEEE Conference on Communication and Network Security, Philadelphia, October 17th – 19th, 2016.

[19] F. Beck and O. Festor. "Syscall interception in xen hypervisor." (2009): 19.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Evaluation-Driven Research in Data Science: Leveraging Cross-Field Methodologies

Bonnie J. Dorr, Peter C. Fontana, Craig S. Greenberg, Marion Le Bras, Mark Przybocki,
National Institute of Standards and Technology
{bonnie.dorr,peter.fontana,craig.greenberg,marion.lebras,mark.przybocki}@nist.gov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

***Abstract*—While prior evaluation methodologies for data-science research have focused on efficient and effective teamwork on independent data science problems *within* given fields [1], this paper argues that an *enriched* notion of evaluation-driven research (EDR) supports methodologies and effective solutions to data-science problems *across* multiple fields. We adopt the view that progress in data-science research is enriched through the examination of a range of problems in many different areas (traffic, healthcare, finance, sports, etc.) and through the development of methodologies and evaluation paradigms that span diverse disciplines, domains, problems, and tasks. A number of questions arise when one considers the multiplicity of data science fields and the potential for cross-disciplinary "sharing" of methodologies, for example: the feasibility of generalizing problems, tasks, and metrics across domains; ground-truth considerations for different types of problems; issues related to data uncertainty in different fields; and the feasibility of enabling cross-field cooperation to encourage diversity of solutions. We posit that addressing the problems inherent in such questions provides a foundation for EDR across diverse fields. We ground our conclusions and insights in a brief preliminary study developed within the Information Access Division of the National Institute of Standards and Technology as a part of a new Data Science Research Program (DSRP). The DSRP focuses on this cross-disciplinary notion of EDR and includes a new Data Science Evaluation series to facilitate research collaboration, to leverage shared technology and infrastructure, and to further build and strengthen the data-science community.**

I. Introduction

The Information Access Division (IAD) of the National Institute of Standards and Technology (NIST) has developed a new Data Science Research Program (DSRP) [2], [3]. While prior evaluation methodologies for data-science research have focused on efficient and effective teamwork on independent data-science problems *within* given fields [1], the DSRP focuses on *enriched* evaluation-driven research (EDR) to support methodologies and effective solutions to data-science problems *across* multiple fields. The motivation for this shift is the need to share common solutions and metrics, while avoiding reformulation of solutions to data-science problems in one discipline that are applicable to problems in another seemingly distinct discipline. The DSRP relies on the development of domain-independent solutions, and thus is well positioned for examination of a range of problems and solutions in many different fields, such as traffic, healthcare, finance, sports, etc.[1]

[1]This paper draws from these four domains for representative examples of data-science solutions; however, many other domains lend themselves to data-science solutions: weather, biology, law, ecology, economics, business, security, medical informatics, social sciences, humanities, and several others.

In this program, data science is viewed as the application of techniques for analysis and extraction of knowledge from potentially massive data. Data science includes notions of big data technical challenges in distributed and parallel processing, as well as considerations and insights that might arise even with smaller datasets. This paper aims to cover methodological questions for data science more broadly, thus subsuming issues inherent in big data challenges.

In this position paper, we adopt the view that progress in data-science research is enabled through this cross-field examination and through the development of methodologies for transferring knowledge of approaches, solutions, measures, and evaluation paradigms across those diverse disciplines, problems, and tasks. Taking this approach yields a significantly enriched notion of EDR beyond that of prior work through the cross-disciplinary sharing of ideas across fields and the discovery of solutions that otherwise would not have been apparent within a given field.

Toward that end, we have developed a new Data Science Evaluation (DSE) series—central to the DSRP—within which evaluations are expected to recur annually. The series consists of several *tracks*, where a track is made up of problems set in a given field of data science. Each track is planned, organized, and implemented by a "track coordinator," either a NIST data scientist or a non-NIST expert in the field of interest.

The first phase of the DSE series was completed in spring 2016: a small-scale pre-pilot evaluation was conducted by NIST that consisted of a single track with a traffic prediction use case in the automotive domain. In the fall of 2016, a pilot evaluation will take place that extends the pre-pilot evaluation track and is open to all who wish to participate. The pilot evaluation is designed to pave the way for a successful and informative full-scale evaluation encompassing multiple disciplines.

Toward that end, the goal for 2017 is to develop an inaugural evaluation, consisting of multiple evaluation tracks in different domains and use cases—championed by experts in different fields. As a first step toward this goal, the upcoming 2016 pilot evaluation encompasses these objectives:

- further develop and exercise the evaluation process at NIST in the context of data science,
- provide participants the opportunity to exercise the evaluation process prior to participating in larger-scale evaluation,

Dorr, Bonnie; Fontana, Peter; Greenberg, Craig; Le Bras, Marion; Przybocki, Mark.
"Evaluation-Driven Research in Data Science: Leveraging Cross-Field Methodologies."
Paper presented at 2nd International Workshop on Methodologies and Tools to improve Big Data Projects,
Washington, DC. December 5, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

TABLE I
FOUNDATIONAL QUESTIONS FOR ENRICHED EDR

A. **Classes of Problems**: What kinds of measurable problems, techniques, and algorithms generalize across domains?
B. **Tasks**: What is an appropriate series of tasks on which data researchers in different fields may want to test their approaches, either at the task-component level or at a higher end-to-end level?
C. **Methods and Metrics**: What kinds of methods and metrics generalize to problems across domains? How effective is the generalization and what can be done to make algorithms and methods more domain-independent?
D. **Ground Truth**: Are there ground-truth considerations common to different areas of data science? Are there effective techniques to handle the lack of or limited ground truth in different both for solving problems and for evaluating methods and metrics?
E. **Data Uncertainty**: What approaches to handling gaps and inconsistencies in data are applicable to multiple domains?
F. **Community Cooperation**: Is it possible to foster cross-field cooperation for sharing of solutions?
G. **Diversity of Solutions**: Can cross-field synergies yield diverse solutions within a given field?

- serve as an archetype for the development of future evaluation tasks, datasets, and metrics,
- establish baseline performance measurements,
- identify new measurement methods and techniques that might be applied to a broad range of use cases, regardless of data type and structure.

It is the last objective above that is the central theme of this position paper. Most notably, although the pilot evaluation is set in the automotive domain, it is expected that many of the algorithms and techniques to be evaluated (as well as the evaluation approaches and metrics themselves) will generalize to other domains. This theme is fundamental to an enhanced notion of EDR where methodologies, approaches, solutions, measures, and evaluation paradigms are considered in new domains, with the expectation that many of these will be of significant utility across diverse fields.

A framework for *enriched* EDR can be characterized as a set of answers to questions such as those shown in Table I. We argue that addressing the issues inherent in such questions provides a foundation for EDR across diverse fields and thus enables research progress on many shared problems and tasks in different domains.

The next section presents related work in evaluation-driven research and generalizability across domains. Following this, Section III makes a case for concrete steps toward this generalizability and explores an enriched notion of EDR through examination of each of the questions above in turn. In Section V, we ground our conclusions and insights in a brief preliminary study developed within the DSE for evaluation-driven research aimed at strengthening the data-science community. Concluding remarks are provided in Section VI.

II. BACKGROUND

The earliest seeds for EDR were sown in the 1970's, when George H. Heilmeier—while serving as Director of the Defense Advanced Research Projects Agency (DARPA)—developed the *Heilmeier Catechism*. (This was published two decades later (as [4]) and was subsequently reviewed by others, e.g., [5].) In this work, several questions related to innovation and novelty were posed, but also a set of evaluation-driven questions were posed along the following lines [4, pg. 15]: "What difference will it make? What are the midterm and final "exams" to check for success?"

These questions are foundational to EDR and have become the basic tenets for many evaluation campaigns across different disciplines. Moreover, these questions have driven progress in many different areas of research. EDR has been around for a long while (more on that below), it has served us well for decades, and it is not likely to go away anytime soon.

The implication is that progress in research is heavily guided by goals related to evaluation. What is often overlooked, however, is an aspect related to the cross-disciplinary (enriched) notion of EDR espoused in this paper. Interestingly, Heilmeier's 1992 article [4] included several important *lessons* that were relevant to this enriched notion, but that (at the time) received less attention than the oft-cited questions in the Heilmeier Catechism. For example, the following statement from the article is relevant [4, pg. 14]: "Approach problems from an interdisciplinary point of view. Remove the barriers to exploiting the viewpoints of other disciplines, and do not be afraid to be called naïve when venturing outside your own professional discipline."

In short, while experts in diverse fields such as those enumerated in Section I might consider their own problems to be unique, it may be the case—more often than not—that data-science solutions and metrics used in one field may be applicable to problems in another field. Before exploring this enriched notion of EDR, it is worthwhile to examine the history leading up to the point where research progress began to be driven in large part by discipline-wide evaluations.

Falling in line with Helmeier, speech researchers experienced a paradigm shift in the 1980's, where evaluations served to push research forward [6], [7]. The DARPA TIPSTER speech evaluation involved evaluation methodologies designed to support focused research, to establish momentum, to maintain continuity, and to encourage longevity, while pushing forward the state of the art. However, evaluation methodologies and metrics spanning domains and disciplines were generally unheard of back in these early days; generally each field had its own evaluation (e.g., speech recognition).

Following this paradigm shift, EDR began to serve as the basis for multiple evaluations by the National Institute for Standards and Technology (NIST), most notably in Information Retrieval, as evaluated in the Text REtrieval Conference [8]. This gave rise to several discipline-wide evaluations with well-defined tasks, data, metrics, and measurement methods (see, e.g., [3] for a more in-depth discussion). Most notably, EDR has successfully spurred research progress in automatic speaker recognition research [9], [10], machine translation [11], and optical character recognition [12].

The associated development of benchmarks has yielded the application of an evaluation methodology to different types of inputs for a given technology. For example, Word Error

Dorr, Bonnie; Fontana, Peter; Greenberg, Craig; Le Bras, Marion; Przybocki, Mark.
"Evaluation-Driven Research in Data Science: Leveraging Cross-Field Methodologies."
Paper presented at 2nd International Workshop on Methodologies and Tools to improve Big Data Projects,
Washington, DC. December 5, 2016 - December 8, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Rate (WER) has been applied to evaluate effectiveness of speech recognition output for varying levels of "informality" (i.e., different genres) of speech inputs [7].[2] However, the application was always "speech transcription" and the metric was always WER. If the application domain were not speech, would this same evaluation paradigm be applicable?

Certainly within the fields of natural language processing and document processing, there are other problems to which WER has been applied: machine translation (often modified to take into account *meaning* such as Human-targeted Translation Error Rate (HTER) [13]) and optical character recognition (often applied in conjunction with character error rate [14]). This is a good start—but what about generalizing to non-language data, such as non-language medical or financial data?

We adopt the view that generalizations of approaches and measures are *often* possible across disciplines upon closer inspection. For example, in many fields, experts are focused on predicting a series of ordered consequences. When an input sequence is mapped into an output sequence for this type of prediction task—regardless of the field—a metric akin to WER could be adapted for a particular problem and, moreover, the underlying technology that implements a given solution might itself be shared across fields. In such cases, the way that the technology is both implemented and evaluated would benefit from cross-disciplinary sharing.

Another clear case of a large-scale data science evaluation that supports EDR is Kaggle (https://www.kaggle.com/)—a forum for hosting Data Science Competitions that consists of a wide range of "challenges" in multiple domains [15]. Although EDR is central to the design of these periodic evaluations, the missing aspect is that of generalizability: each competition is run (mostly) independently, with little or no synergy across the different problem areas, tasks, and and also no shared methodologies (unless by accident). Kaggle does include a means for employing metrics across domains, which is an important step toward generalizability; our aim is to further enrich the notion of EDR through cross-domain sharing of algorithms for analogous tasks.

The CLEF Initiative (Conference and Labs of the Evaluation Forum, formerly known as Cross-Language Evaluation Forum) [16] is yet another example of a large-scale evaluation that is aligned with EDR. The main mission of CLEF is to promote research, innovation, and development of information access systems with an emphasis on multilingual and multimodal information with various levels of structure. CLEF has had a very significant scholarly impact, as evidenced by the emergence of research directions that otherwise would not have been possible and also by measures of the scholarly impact of the research fostered by benchmarking activities within CLEF.

The overarching field of study for participants in CLEF is information retrieval, but the application areas are often broad, encompassing numerous disciplines. As a forum that supports "Evaluation Campaigns" CLEF has paved the way for providing a basis for multi-disciplinary sharing that might be leveraged for an enriched notion of EDR as described in this paper. Concrete steps toward bringing about sharing of techniques and metrics—such as those discussed in the next section—might improve progress on CLEF problems both within and across evaluation campaigns.

Several other evaluation-related concepts and paradigms are related to EDR and, thus, would potentially benefit from multidisciplinary participation and enriched EDR. For example, Developmental Evaluation (DE) [17] leverages evaluation to support development and to focus on long-term, continuous improvement. This domain-independent methodology applies to many fields. Relatedly, in software engineering, the concept of a *benchmark*—a common framework for people within a discipline to discuss and compare solutions—can provide EDR benefits. Sim et al. [18] argue that benchmarking advances research by providing a setting where researchers can focus their attention on key problems. Sharing of benchmarking approaches across disciplines provides another opportunity to generalize evaluation methodologies in ways that may lead to EDR enrichment. One such framework is the use of common data sets to compare systems and algorithms. SPEC [19] and the UCI Machine learning repository [20] are two such collections of data sets.

The next section discusses concrete steps toward generalizability of concepts and paradigms related to evaluation and further develops the notion of enriched EDR.

III. An Enriched Notion of Evaluation-Driven Research for Data Science

Prior work [3] conceives of EDR for data science as a notion that is divided up into four steps that are more programmatic than foundational in nature: planning tasks and research objectives for evaluation; design of data and experiments; performance assessment; and provision of a forum to discuss evaluation outcomes. These steps are crucial to the development of a framework within which to *practice* EDR, but they do not translate directly into the fundamentals of how to *implement* various aspects pertaining to EDR. For example, these programmatic steps remain agnostic with respect to how one studies different problems, tasks, and domains in data science in a way that leverages evaluation methodologies that might potentially span multiple disciplines.

Enriched EDR is achieved when a solution in an existing field is applied successfully in a new field for which such a solution had not been previously imagined. As alluded to above, data science crucially spans a diverse set of disciplines and domains [21]. Leveraging the breadth of disciplines investigated by data scientists provides the basis for strengthening EDR to the advantage of each of the individual fields.

To date, EDR has driven progress with an evaluation cycle that is repeated at regular intervals and, as technology improves, the research objectives are made more challenging on each subsequent cycle. Within data science, this repeated cycle has the potential for further enrichment through the ability to

[2] In addition to level of informality, speech evaluation includes other speech-specific dimensions such as the language, noise, microphone type, and environment, each increasing the difficulty of the same challenge over time.

Dorr, Bonnie; Fontana, Peter; Greenberg, Craig; Le Bras, Marion; Przybocki, Mark.
"Evaluation-Driven Research in Data Science: Leveraging Cross-Field Methodologies."
Paper presented at 2nd International Workshop on Methodologies and Tools to improve Big Data Projects,
Washington, DC. December 5, 2016 - December 8, 2016.

TABLE II
REPRESENTATIVE SET OF DOMAIN-SPECIFIC TASKS FOR FOUR DOMAINS (TRAFFIC, HEALTHCARE, FINANCE, AND SPORTS) ACROSS THREE DATA-SCIENCE PROBLEMS: ANOMALY DETECTION, PREDICTION, AND ALIGNMENT

		Problem		
		Anomaly Detection	**Prediction**	**Alignment**
Domain	**Traffic**	Cleaning up gaps and inconsistencies in lane detector data	Determining upcoming traffic speed using flow volume and percentage occupancy	Relating traffic events in reports to traffic video segments containing those events
	Healthcare	Detecting outliers for discovery of fraudulent claims	Predicting patients' next-day care needs from electronic patient records	Correlating patient data for patients across multiple medical databases
	Finance	Detecting potential change in direction and momentum of market	Identifying potential stock market events from sentiments in social media	Aligning past news reports with previous stock market events
	Sports	Detection of anomalies in athletic performance data (e.g., injuries, doping)	Predicting successful athletic strategies (i.e., when to pull out and/or substitute a player in a team sport)	Identifying athletes across multiple performances (i.e., games or competitions)

leverage tasks, methodologies, and metrics across fields that might at first glance appear to be too distinct for such sharing.

IV. DISCUSSING A SOLID FOUNDATION FOR EDR

This section highlights the potential for transferring knowledge of approaches, solutions, measures, and evaluation paradigms across those diverse disciplines, domains, problems, and tasks. Throughout this discussion, the goal is to determine what it would take to establish a solid foundation for enriched EDR and to provide data scientists easy access to methodologies, solutions, and metrics from multiple fields. The aim is to enable the discovery of new ways to use these that were not previously anticipated, thus enabling forward progress on longstanding research problems across multiple fields.

Each of the questions presented in Table I will be addressed in turn, below. We adopt the view that addressing the issues inherent in such questions provides a foundation for an enriched notion of EDR that spans a range of diverse fields.

A. Classes of Problems: What kinds of measurable problems, techniques, and algorithms generalize across domains?

Identifying the classes of measurable problems, techniques, and algorithms that generalize across data-science domains is crucial for an enriched notion of EDR. In the work of [22], classes of problems in data science are outlined, with four example case studies that span these classes. The disciplinary foundation for these four studies is the same: language processing (topic modeling, emotion-word mining, and informal language analysis). The primary problem of interest in this work is classification of vocabulary usage in news and blog data.

There are, however, a number of problems in data science that span a diverse range of fields and that serve as the basis of EDR. Consider the following set of data-science problems, and their application to a representative set of domains, as shown in Table II:[3]

- Anomaly Detection: identification of items or events that do not conform to an expected pattern
- Prediction: estimation of variables of interest at future times
- Alignment: correlation of different instances of the same object

Although these three problems are generalizable across domains, the specific contexts for each of these problems varies across diverse fields—as illustrated by the range of different tasks under each of the problem headings in Table II. The next section discusses the nature of such tasks, after which we argue that—despite domain-specific contexts surrounding these tasks—there are domain-independent solutions and metrics for data-science problems that enable generalizability, thus enriching EDR.

B. Tasks: What is an appropriate series of tasks for which data researchers in different fields may want to test their approaches, either at the task-component level or at a higher end-to-end level?

It is a significant undertaking to find an appropriate series of tasks for which data researchers in different fields may want to test their approaches. Among other challenges, to make progress in EDR, the tasks that are selected for an evaluation must be both technically challenging and a significant step beyond current state of the art.

A comparison of solutions is a central component of all tracks in a community-wide evaluation, and such comparisons often serve as the basis of support for future research. A wide evaluation may be used to determine whether there exist solutions that were not widely known in a given discipline at the time a problem was posed—or it may bring about new solutions that otherwise never had been considered. Scoping out the space of possible solutions (whether pre-existing or proposed anew) is critical for moving forward in ways that enable the development of approaches that are revolutionary, not evolutionary.

To support this endeavor, it is critical to ensure a low enough "barrier to admission" so that the range of tasks selected for evaluation are addressable by researchers across disciplines (hence diverse solutions), while bringing together a broader, international community of data-science researchers. This allows different approaches to be compared and the leveraging of cross-domain synergies.

[3]Additional classes of problems might be considered for examination of methodologies and EDR across domains [3]. For brevity and illustrational clarity, only four are presented here.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

TABLE III
POSSIBLE DOMAIN-INDEPENDENT METHODS AND METRICS TO DATA-SCIENCE PROBLEMS. NOTE THAT THE METRICS FOR PREDICTION ASSUME A CONTEXT WHERE A CONTINUOUS VALUE (AND NOT A PROBABILITY) ARE BEING PREDICTED.

Problems	Methods	Metrics
Anomaly Detection	Timeseries outlier detection, Statistical deviation detection	Accuracy, precision, recall, F1-Score, ROC or DET area, decision cost function, and average precision
Prediction	Multiple regression, Random forests of regression trees	Mean absolute error, Mean squared error, Root mean squared error, R^2 (a correlation metric).
Alignment	Substring matching, hidden markov models	Same metrics as for anomaly detection.

A series of tasks for which data researchers in different fields can build solutions that are evaluated—both at the component level and at the end-to-end level—is important for taking EDR to the next level beyond existing approaches to evaluation. Some examples of domain-specific tasks are enumerated under each three problems shown in Table II.

What is interesting to note is that, despite the domain specificity of tasks, the solutions fall under the heading of much broader categories of problems in data science. As such, specific tasks such as "identifying errors" and "detecting change in direction" fall under the general problem of Anomaly Detection, "predicting care needs" and "identifying future stock market events" fall under the general problem of Prediction, and "relating traffic events (in reports and video)" and "identifying athletes across performances" fall under the general problem of Alignment. Thus, it is expected that similar—or possibly the same—methods and metrics would be applicable for each of these generalized problems, even in cases where the specific tasks seem to be entirely different at first glance. (This point will be discussed further in the "Methods and Metrics" section below.)

Additionally, different tasks need to be combined and evaluated in a workflow in order to better evaluate the consequences of error in one task on another. For instance, different tasks are pipelined, as one example, as five stages in the "Big Data pipeline" [23] described in Jagadish et al. [23]: data acquisition; information extraction and cleaning; data integration, aggregation, and representation; modeling and analysis. By choosing tasks that complete a workflow, such as the pipeline above, individual component-level evaluations can be combined to better evaluate the data analytic process as a whole.

C. Methods and Metrics: What kinds of methods and metrics generalize to problems across domains? How effective is the generalization and what can be done to make algorithms and methods more domain-independent?

One goal of this application of EDR to data science is to develop enhanced methods and metrics that are general-purpose and can be applied to different data analytic components in a variety of domains. However, given the details of the different domains, this can be challenging.

Consider the range of domain-specific tasks associated with each of the three problems presented earlier in Table II. Some possible domain-independent methods and metrics for these tasks are summarized in Table III.

Even if the notion of *anomaly* is clearly specified for a given domain (which may not be the case), the methods associated with this notion may differ based on the context. In many cases (such as in the traffic case), anomalies are viewed as potential errors to be cleaned, yet in other cases (such as in the healthcare domain), an anomaly is considered an important point of interest; indeed, it might be the key to detecting critical information such as fraudulent claims. As such, the methods associated with anomaly detection differ depending on the use case, and the metrics would reward methods in different ways.

However, despite such distinctions, EDR is likely to be enriched by anomaly-detection techniques that work well in one domain and are successfully applied in another domain—even if the results are handled differently by downstream processes. When methods in one domain are "borrowed" for another domain, various forms of adaptation may be needed for maximum efficacy. For instance, in the traffic case the data might be viewed as time-series data where the local (timewise) points might be leveraged to assist in cleaning (Basu and Meckesheimer [24] provide one such algorithm), whereas in the healthcare case time-locality may not be available or relevant. Adaptation of existing techniques—or combinations of cross-field techniques—may yield a solution for both disciplines that might not otherwise have been considered.

As for generalization of metrics to problems across domains, consider the case of Prediction from Table II. If a continuous answer is needed, e.g., "determining upcoming traffic speed," there are a variety of metrics including mean absolute error , mean squared error, root mean squared error, and correlation metrics, such as R^2. If a discrete value is required for a prediction task, e.g., "predicting successful athletic strategies," there are also a variety of metrics that would be applicable across domains, including accuracy, precision, recall, F1-score, Receiver Operating Characteristic (ROC) or Detection Error Tradeoff (DET) curve, and average precision. Although it is often the case that certain metrics are traditionally used within certain disciplines or domains, there is no reason to expect that such metrics would not be applicable to multiple disciplines or domains—thus yielding insights that enable research progress in individual disciplines.

Caruana and Niculescu-Mizil [25] showed that supervised learning algorithms may vary heavily depending on the metric used, and it is often not clear which metric is most relevant for the specific domain. Even if a metric is general-purpose, the contexts in which it matters may not be. One approach

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

to designing a general-purpose metric that can be applied in a variety of domains is to simulate (through different datasets) how the score of each metric varies in different situations. Another approach is to pick metrics that have desirable properties, such as symmetry. Xiong and Li [26] performed a study on various clustering metrics to illustrate different properties of those metrics and to single out metrics that may exhibit desirable properties in a variety of scenarios and datasets.

The idea of understanding contexts where certain methods score lower according to certain metrics is a part of EDR; likewise, the idea of understanding the strengths, limitations, and properties of metrics is a key component of EDR. Evaluations that apply different metrics to compare different methods provide insight into general-purpose methods and metrics, allowing comparisons to be made with respect to the generalizability of the metric to different contexts.

D. Ground Truth: Are there ground-truth considerations common to different areas of data science? Are there effective techniques to handle the lack of ground truth in different domains, both for solving problems and for evaluating methods and metrics?

The ability to obtain and use ground truth is fundamental to measuring the effectiveness of research, both within and across domains. Although the existence of ground truth is an inescapable part of almost any data-science endeavor, identifying it may require a significant effort and is often fraught with challenges and limitations, regardless of the domain.

At the source of many recurring problems associated with data-science evaluation—regardless of domain—is the absence of the "right" data against which to evaluate.[4] For example, there may be no way to collect the data in the first place, or the data collected may not be varied enough to represent the range of cases that actually arise, or the data collected "after the fact" may be unrealistic in some way that is a mismatch with the actual situation. More concretely, the following are the difficulties that are likely to occur in many different areas of data science:

- The "answer" to the problem could be unknown, i.e., *ground truth is completely missing*. This could be because humans cannot know the answer—the information does not exist (e.g., predicting a system output that will compel a user to take a particular action). Alternatively, if the information does theoretically exist, it may by design or otherwise never be recorded (e.g., grouping bitcoin wallets). Having no "answer" for comparison of system outputs means no comparative analysis of accuracy can be made. In such cases, determining whether a system actually addresses the problem it set out to solve becomes a significant challenge. However, should the problem fall into a well defined category of data science, mapping the system to another domain may give an estimation of how well the system addresses the given problem. For example, if ground truth is unavailable to evaluate patient matching algorithms (matching different occurrences of the same patient across data sources, e.g. hospitals, GPs, labs etc., where there can be many data-entry mistakes), researchers may try to evaluate the same algorithm in the domain of athlete matching, where information is less sensitive and therefore more available.[5]
- Ground truth does exist but may have significant limitations (e.g., only 1% of the data are labeled). This could be for multiple reasons, often due to the expense of gathering ground truth, whether in terms of resources, machine time, or person hours. Evaluating on limited ground truth introduces biases that hinder the ability of researchers to correctly assess the accuracy of their systems. For example, in a classification problem, or some classes could be missing, the spread of classes could be unrepresentative of the main dataset. Techniques to limit these biases have been developed. For example, Katariya et al.'s [27] work on active evaluation of classifiers across multiple domains estimates accuracy by utilizing a (usually human) labeler to construct a small (adaptive) labeled set, which is then used as limited ground truth.
- Ground truth does exist but is either partial or unreliable. In the case of partial ground truth, the answers may be available only under specific conditions or they may represent only part of the relevant information. In the case of unreliable ground truth, the answers may be associated with low-confidence output, e.g., results obtained from crowd-sourcing approaches such as "turking".[6] Being able to use partial or unreliable answers is sometimes the only way to evaluate systems for a given problem, given the lack of complete and/or accurate "answers." Previous NIST evaluations (e.g., [8]) apply accuracy measures that accommodate the lack of full truth data, often employing mediated adjudication approaches (e.g., pooling relevance assessments of participants in the evaluation to approximate recall).

Therefore with respect to ground truth, we make two claims: (1) The more standardized the tasks, measurement methods, and metrics, the easier it is to adapt techniques across problems (see the above example in the case of absent ground truth); (2) Techniques that are developed in one domain to address incompleteness of ground truth can be adapted to other domains.

E. Data Uncertainty: What approaches to handling gaps and inconsistencies in data are applicable to multiple domains?

Uncertainty arises in every domain. It comes from measurement error and noise that add variance and, possibly,

[4] In this paper, Ground Truth is referred to in the context of the evaluation (i.e. "test") side, but it is worth noting that methods that address the lack of, or limited, Ground Truth may also significantly aid training purposes.

[5] Of course, mapping a problem to a different domain gives rise to its own set of problems (e.g., there may be a different distribution of errors for athlete matching than for patient matching), and this may impact effectiveness of systems as well.

[6] Using frameworks like Amazon Mechanical Turk [28], researchers can outsource the labeling of their data. This method is often considered unreliable as workers are paid by the amount of data labeled and are anonymous.

TABLE IV
EVALUATION METHODOLOGY: DOMAIN-INDEPENDENT METRICS TO BE APPLIED IN THE PILOT EVALUATION (FALL 2016)

Data Science Problem (Task)	Measures of Success (Metric)
Cleaning$_{Det}$ and Alignment	Decision Cost Function (DCF) representing a linear combination of the miss and false alarm rates at a threshold τ. The overall performance metric is the minimum DCF value obtained considering all τ: $\min_\tau (DCF(\tau))$, where $DCF(\tau) = c_{miss} * P_{target} * \frac{\lvert\text{misses}(\tau)\rvert}{\lvert\text{target trials}\rvert} + c_{fa} * (1 - P_{target}) * \frac{\lvert\text{false alarms}(\tau)\rvert}{\lvert\text{non-target trials}\rvert}$
Cleaning$_{Cor}$	Mean Absolute Error (MAE), where n is the number of trials, and for each trial i: $\widehat{v_i}$ is the estimated data value and v_i is the correct data value. The overall performance metric is: $MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert\widehat{v_i} - v_i\rvert$
Prediction	Root Mean Squared Error ($RMSE$), where $\mathcal{E}$ is the set of event types, n is the number of trials, and for each trial i, $\widehat{e_i}$ is the predicted count of events of type e and e_i is the true count of events of that type. The overall performance metric is the average of each trial's $RMSE$: $\frac{1}{n}\sum_{i=1}^{n} RMSE(i)$, where $RMSE(i) = \sqrt{\frac{1}{\lvert\mathcal{E}\rvert}\sum_{e\in\mathcal{E}} (\widehat{e_i} - e_i)^2}$

bias to the measured value. To understand the nature of uncertainty in a more general context, consider, for example, multiple measurements of an object by hand with a ruler: it is expected that slightly different lengths would be obtained for each measurement. The resulting distribution of measurement values provides a representation of the uncertainty.

Despite uncertainty's impact on measurement, it is common to ignore uncertainty at one or more stages of the data-to-decision pipeline. In the worst case, uncertainty cascades, causing the output to have no positive bearing on the decision.

Such concerns can be addressed by overcoming several challenges associated with the handling of uncertainty, namely how to represent it, how to communicate it, how to measure it, and how to calibrate its measurement—all at both the component and pipeline levels. We adopt the view that a cross-field analysis supports the development of methodologies for measurement and handling of uncertainty across all domains.

Advances in measuring uncertainty in computer simulations has impacted meteorology [29], medicine [30], and many other domains [31]. Similarly, progress in handling uncertainty in, for example, clustering could impact every domain in which challenges naturally modeled as clustering problems arise, such as image segmentation (where image pixels are grouped by the object the represent) and entity resolution (where mentions of entities in text are grouped by the real-life entity to which they refer).

Thus the potential impact of progress in handling uncertainty in data science is exceptionally broad and its importance will continue to rise as system performance improves to the point where uncertainty is large relative to the performance error[7] and as data science is more commonly used to discover knowledge and make decisions of great consequence.

F. Community Cooperation: Is it possible to foster cross-field cooperation for sharing of solutions?

Even in the days of TIPSTER [6], evaluation frameworks and their accompanying infrastructure were designed to foster collaboration and progress. However, such programs were focused on problems within one discipline (speech recognition in the case of TIPSTER), in contrast to the data science arena, which is expected to bring together vastly different disciplines.

[7]which, incidentally, is when one might most want to use a system and to understand the uncertainty of a given output.

Moreover, in many of the early, single-discipline evaluations (such as *machine translation* [11]), the goal was to find solutions that achieved *as close to* human performance as possible. It could be argued that this goal is entirely supplanted by a very different one in the field of data science, which is to *surpass* human performance (as data sizes are often far too large for humans to achieve the same degree of speed or to manage the cognitive load and fatigue that lead to human error) while *not reinventing* the wheel (as the solution to a data-science problem in one field may have already been found in another). Both goals are difficult to achieve, but they are quite different. For the latter, it is clear that a more concerted effort is needed to encourage cross-fertilization, with novel paradigms that foster collaboration in some very new ways.

Data science problems can be viewed through a much broader lens than that of many prior evaluations. To ensure that researchers in one field have access to knowledge about problems and solutions in another field, it is important that the *evaluation methodology itself* be designed to span multiple domains. In so doing, problems may, in fact, be shared across entirely independent tracks within a data-science evaluation series, and the evaluation framework thus enables accessibility to solutions and metrics that span multiple domains.

At first it may seem difficult—or impossible—for such a large of researchers who study such a vast array of disciplines (such as those from Table II) to come together in a way that enables transfer of knowledge of approaches, solutions, measures, and evaluation paradigms. However, it has become clear that this is absolutely essential with large data sets across many domains, and the potential for sharing common solutions to managing such large sets.

Researchers are willing, more now than ever before, to openly discuss their successes and failures in the context of data-science evaluation. As community-wide evaluation fora (e.g., NIST, Kaggle, and CLEF) and multi-week research workshops (e.g., Frederick Jelinek Memorial Workshops on Speech, Language and Computer Vision [32]) have become increasingly grounded in multi-disciplinary problems, opportunities have emerged to leverage the openness and community spirit that is necessary for an enriched notion of EDR. This type of enrichment was elusive in the early days of EDR, when researchers and companies were expected to compete for status or top ratings—academics climbing the tenure ladder, researchers battling it out for recognition through awards and

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

funding, and commercial entities holding their "secret sauce" under lock and key.

While some of these competitive aspects are still present nowadays, many communities of researchers are now willing to share ideas, approaches, and lessons learned—and to collaborate in ways that yield better results than would be achieved individually. The time is ripe to take advantage of this forward motion and apply this collaborative spirit not just *within* research communities, but *across* them. The field of data science is uniquely positioned, with its inherent multidisciplinarity, to provide the basis for an enriched EDR at a macro level that has not previously been attained.

A concrete example of an evaluation methodology that can enhance both within- and across-discipline cooperation is the provision of a solid evaluation infrastructure for *Evaluation-as-a-Service* (EaaS) [33], which allows participants to submit systems (through a software container) directly to the evaluator who runs the systems on the evaluator's computing resources and (sometimes hidden) data. Although EaaS sometimes prevents data sharing, it broadens the range of disciplines within which problems can be studied by enabling evaluations even in contexts where the data are sensitive, private, or cannot be shared for other reasons. This paradigm is a direct contrast to those where participants run their methods on their computing resources and then submit results to the evaluator for scoring. A strong case can be made for EaaS as an evaluation methodology that scales to the size and complexity of data found in many fields of data science (e.g., medical data sets). Such an approach is an initial step toward the notion of enriched EDR presented in this paper.

G. Diversity of Solutions: Can cross-field synergies yield diverse solutions within a given field?

For any field, independent solutions may yield rather large gains when combined, if the independent solutions are based on approaches that are diverse enough to yield significant complementarity. Researchers who espouse "hybrid approaches" rely heavily on very distinct sub-approaches such that the combination yields the highest performance gains. Many examples are seen in the field of language, where statistical approaches are combined with symbolic approaches [34] or in tasks related to autonomy and coactive design where automatic techniques are combined with human-in-the-loop guidance [35].

Data science is inherently diverse due to the degree of multidisciplinarity. The solutions expected from data-science researchers are likely to be applicable across different domains—yet be distinct enough to be complementary, such that combinations of approaches may be generated and tested within the evaluation paradigm described in this paper. For example, dynamic programming algorithms used for efficient alignment of data sequences have potential to be applied to alignment tasks in both the traffic and healthcare domains.

An even more compelling result—leading to enriched EDR—is one where an algorithm from one field is combined with that of another field to yield improvements over either one alone. For example, *hidden markov models*, which have been used in conjunction with dynamic programming for alignment problems in biological and healthcare domains for years (e.g., protein sequence alignment [36]), might be used in conjunction with dynamic programming in the traffic domain to improve performance of systems that align traffic video segments to traffic incident reports.

An important enabler of diversity and cross-field synergies (and thus diverse solutions) is global accessibility of evaluation resources and infrastructure. One approach that successfully involves geographically diverse participants is that of Kaggle [15], for which participants invest hundreds of hours in exploring the potential solution space, as mentioned above in Section II. Kaggle competitions include a large number of well-developed tutorials that have been designed to lower the barrier to admission, and solutions to Kaggle competitions are compared within specific problems.

The degree to which cross-domain synergies are leveraged for solutions across different data-science problems (e.g., development of alignment techniques for problems in both finance and sports) is expected to be higher within an evaluation framework that supports a unified effort that covers many different areas of data science simultaneously. Toward that end, we have conducted a case study based on an upcoming pilot evaluation of traffic incident detection and prediction. The goal is to solidify a framework that is designed to support enriched EDR for multi-track data-science evaluation in future years.

V. A Case Study: NIST's Experience with Traffic Incident Detection and Prediction

In formulating the problems to be addressed for the upcoming (Fall 2016) Pilot evaluation, the NIST team has adhered to the guiding principle underlying the Data Science Evaluation series that the tasks to be evaluated would span multiple fields, such that researchers would be able to test their approaches across different evaluation tracks (domains)—thus giving rise to an enriched EDR and accelerated research progress.

The initial tasks designed for this Pilot Evaluation fall within the realm of the three data-science problems below, with specific application to the traffic domain as an example of each case.

1) **Cleaning.** Detecting and correcting errors in traffic lane detector data flow values.
2) **Alignment.** Matching traffic events with traffic video segments containing those events.
3) **Prediction.** Inferring the number and types of traffic incidents in an upcoming time interval based on historical data.

A fourth data-science problem related to prediction is also included in the Pilot evaluation but not highlighted in this paper is *forecasting*, which produces a timeseries for the predicted values. Like the other three, this task has an analogous standing in other domains such as patient matching in healthcare records, financial trending, and sports.

For this case study, we successfully arrived at an evaluation methodology that included measures of success. We apply this methodology for the Pilot evaluation using the specific

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

metrics and tasks shown in Table IV to the traffic domain, where each metric in the table is the metric used to measure a system's performance on the specific data analytic task. For example, misses and false alarms (for Cleaning$_{Det}$ and Prediction) are defined in terms of traffic events, data values (for Cleaning$_{Cor}$) are defined in terms of traffic flow, and event types (for Alignment) are defined in terms of a range of different types of traffic incidents.

Note that the *detection* task associated with Cleaning—a variant of the "Anomaly Detection" problem—is evaluated by the same metric that is used for Alignment (the Decision Cost Function). However, there is a second task associated with Cleaning, *correction*, that is evaluated by an entirely different metric: Mean Absolute Error (MAE). Prediction is evaluated using yet another metric, Root Mean Squared Error (RMSE).

We further note that there are subtle differences between the metric for the detection task (for the Cleaning problem) and the metric for the Alignment task. The inclusion of the parameters c_{miss}, c_{fa}, and P_{target} allow the metric to be customized so that it is applicable to a variety of scenarios. For the traffic use case, the parameter values selected for this decision cost function are based on two different scenarios: for detection, $c_{miss} = 1$, $c_{fa} = 1$, and $P_{target} = 0.0312$; for alignment, $c_s = 1$, $c_{fa} = 100$, and $P_{target} = 0.5$.

This *within-domain* parameterizability provides a level of flexibility for the specification of metrics that also enables applicability *across* domains, in support of enriched EDR. For example, the evaluation of anomaly detection in financial trending would be analogous to that of the detection task for data cleaning in the traffic domain, where misses and false alarms in the Decision Cost Function would be re-mapped from the typical traffic events (such as an accident) to financial events (such as a major change in stock price). Similarly, patient matching in the healthcare domain could be viewed as an Alignment task, where the problem of correlating patient data for patients across multiple medical databases is treated analogously to matching traffic events with traffic video segments containing those events; the same Decision Cost Function metric used in the traffic domain would then be applicable to the healthcare domain for this problem.

Data uncertainty received limited attention in the pilot, though arose in the form of noise and error present in the traffic sensors. These were handled by deleting obviously errorful measurement values, and otherwise treating the sensor output as accurate. In this sense, the cleaning task is focused on uncertain data. Incorporating uncertainty into system output for NIST to measure is a likely focus of future work in data uncertainty.

Challenges with respect to ground truth in the pilot were addressed by carefully designing tasks, "removing data", and through selective annotation. It is worth noting that this is an evaluation problem, not entirely dissimilar to the problem of unsupervised (or semi-supervised or active) learning.

The pilot evaluation will prototype the EaaS concept described earlier [33] by testing the concept of accepting systems as submissions. Although not previously referred to as EaaS, prior NIST evaluations (including the Fingerprint Vendor Test Evaluation) have utilized this framework of accepting algorithms as submissions and running them internally to evaluate them. The DSE pilot includes this in-house running and evaluation of algorithms—an approach that brings the experiment to the data (rather than the other way around) and that enables additional evaluation features including system performance benchmarking.

Finally, it is already the case that cross-field cooperation has played a central role in the development of the data-science evaluation series, with an enriched EDR at the heart of its design. In the very first workshop on this new evaluation series [37], attendees expressed interest in forming new tracks. Two proposals were presented in the areas of plant identification and predictive security analytics. It is expected that there will be 2–3 new tracks within the next year, on problems analogous to those shown in Table II, and that research will progress effectively and efficiently due to the sharing of algorithms and their combinations.

VI. Conclusion

We have argued that an enriched notion of *Evaluation-Driven Research* (EDR) supports methodologies and effective solutions to data-science problems across multiple fields. We have provided a methodology and evaluation design within which progress in data-science research is enabled through access to techniques that are applicable to, and valuable for, problems in different disciplines.

This paper espouses the view that, to ensure success of this enriched EDR paradigm, it is important to examine challenges associated with cross-field generalizations, and to invest effort and time in adapting work that has already been done by researchers in different fields.

We have grounded our conclusions and insights in a brief preliminary study as a part of a new Data Science Research Program (DSRP). We have defined a set of goals within this program that enables the building and strengthening of different areas of evaluation within data science. As a part of this program, we have built a new Data Science Evaluation (DSE) series in which the principles spelled out in this position paper are currently being leveraged for a multi-track evaluation series with broad coverage of problems in diverse fields.

Most notably, we have made the case that, through cross-field generalizations of tasks, measurement methods, and metrics: (1) solutions in existing fields may be applied successfully in a field for which such solutions had not been previously imagined; (2) techniques that are developed in one domain for understanding and representing uncertainty and addressing ground-truth considerations can be applied to another domain for which such techniques have not yet been discovered; and (3) cross-domain synergies may be leveraged in an evaluation framework that supports a unified effort that covers many different areas of data science simultaneously.

Disclaimer

These results are not to be construed or represented as endorsements of any participants system, methods, or com-

mercial product, or as official findings on the part of NIST or the U.S. Government.

Certain commercial equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the equipment, instruments, software or materials are necessarily the best available for the purpose.

REFERENCES

[1] J. S. Saltz, "The need for new processes, methodologies and tools to support big data teams and improve big data project effectiveness," in *2015 IEEE International Conference on Big Data (Big Data)*, Oct. 2015, pp. 2066–2071.

[2] B. Dorr, C. Greenberg, P. Fontana, M. Przybocki, M. Le Bras, C. Ploehn, O. Aulov, M. Michel, E. J. Golden, and W. Chang, "The nist iad data science research program," in *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015, pp. 1–10.

[3] B. J. Dorr, C. S. Greenberg, P. Fontana, M. Przybocki, M. Le Bras, C. Ploehn, O. Aulov, M. Michel, E. J. Golden, and W. Chang, "A new data science research program," *International Journal of Data Science and Analytics*, vol. 1, no. 3, 2016.

[4] G. Heilmeier, "Some reflections on innovation and invention," in *Founders Award Lecture, National Academy of Engineering*, Washington, D.C., 1992. [Online]. Available: https://www.isi.edu/~johnh/TEACHING/CS651/ARCHIVE/Heilmeier92a.pdf

[5] J. Shapiro, "George h. heilmeier," *IEEE Spectrum*, vol. 31, no. 6, pp. 56–59, 1994. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=284787

[6] J. D. Prange, "Evaluation Driven Research: The Foundation of the TIPSTER Text Program," in *Proceedings of a Workshop on Held at Vienna, Virginia: May 6-8, 1996*, ser. TIPSTER '96. Stroudsburg, PA, USA: Association for Computational Linguistics, 1996, pp. 13–22. [Online]. Available: http://dx.doi.org/10.3115/1119018.1119022

[7] D. S. Pallett, "A look at NIST's benchmark ASR tests: past, present, and future," in *ASUR 2003: IEEE Workshop on Automatic Speech Recognition and Understanding, 2003.* IEEE, 2003, pp. 483–488. [Online]. Available: http://itl.nist.gov/iad/mig/tests/rt/ASRhistory/pdf/NIST_benchmark_ASRtests_2003.pdf

[8] "Text retrieval conference," 2014. [Online]. Available: http://trec.nist.gov

[9] D. Reynolds, "Speaker and language recognition: A guided safari," 1 2008, keynote speech at Odyssey 2008. [Accessed: 2015 07 15].

[10] M. Przybocki and A. Martin, "NIST speaker recognition evaluation chronicles," *Computer Speech & Language*, vol. 20, no. 23, pp. 15–22, Apr. 2006.

[11] "NIST open machine translation evaluation," 2015. [Online]. Available: http://nist.gov/itl/iad/mig/openmt15.cfm

[12] "NIST Open Handwriting Recognition and Translation Evaluation (OpenHaRT)," 2010. [Online]. Available: http://www.nist.gov/itl/iad/mig/hart.cfm

[13] M. Snover, N. Madnani, B. J. Dorr, and R. Schwartz, "Fluency, adequacy, or hter?: Exploring different human judgments with a tunable mt metric," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*, ser. StatMT '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 259–268. [Online]. Available: http://dl.acm.org/citation.cfm?id=1626431.1626480

[14] O. Kolak, W. Byrne, and P. Resnik, "A generative probabilistic ocr model for nlp applications," in *Proceedings of the HLT-NAACL*, 2003, pp. 55–62. [Online]. Available: http://www.aclweb.org/anthology/N03-1018

[15] R. Metz, "Startup turns data crunching into a high-stakes sport," *MIT Technology Review*, vol. Spring, 2012. [Online]. Available: https://www.technologyreview.com/s/426796/startup-turns-data-crunching-into-a-high-stakes-sport/

[16] T. Tsikrika, B. Larsen, H. Müller, S. Endrullis, and E. Rahm, "The scholarly impact of clef (2000–2009)," in *Lecture Notes in Computer Science: Information Access Evaluation. Multilinguality, Multimodality, and Visualization (Volume 8138)*, 2009, vol. 8138.

[17] M. Q. Patton, "Developmental evaluation," *Evaluation Practice*, vol. 15, no. 3, pp. 311–319, Oct. 1994.

[18] S. E. Sim, S. Easterbrook, and R. C. Holt, "Using Benchmarking to Advance Research: A Challenge to Software Engineering," in *Proceedings of the 25th International Conference on Software Engineering*, ser. ICSE '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 74–83. [Online]. Available: http://dl.acm.org/citation.cfm?id=776816.776826

[19] "Standard Performance Evaluation Corporation (SPEC)," Website, 2016. [Online]. Available: https://www.spec.org/

[20] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[21] J. Leek, "The key word in data science is not data, it is science." *Simply Statistics*, 2013. [Online]. Available: http://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/

[22] M. Das, R. Cui, D. R. Campbell, and R. R. Gagan Agrawal, "Towards methods for systematic research on big data," in *2015 IEEE International Conference on Big Data (Big Data)*, Oct. 2015, pp. 2072–2081.

[23] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big Data and Its Technical Challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2611567

[24] S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: an application to sensor data," *Knowledge and Information Systems*, vol. 11, no. 2, pp. 137–154, Aug. 2006.

[25] R. Caruana and A. Niculescu-Mizil, "An Empirical Comparison of Supervised Learning Algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 161–168. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143865

[26] H. Xiong and Z. Li, "Clustering Validation Measures," in *Data Clustering: Algorithms and Applications*, C. C. Aggarwal and C. K. Reddy, Eds. CRC Press, 2013, ch. 23, pp. 571–605.

[27] N. Katariya, A. Iyer, and S. Sarawagi, "Active evaluation of classifiers on large datasets," in *2013 IEEE 13th International Conference on Data Mining*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2012, pp. 329–338.

[28] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspectives on Psychological Science*, vol. 6, pp. 3–5, 2011.

[29] J. M. Murphy, D. M. Sexton, D. N. Barnett, G. S. Jones, M. J. Webb, M. Collins, and D. A. Stainforth, "Quantification of modelling uncertainties in a large ensemble of climate change simulations," *Nature*, vol. 430, no. 7001, pp. 768–772, 2004.

[30] R. S. Dittus, S. D. Roberts, and J. R. Wilson, "Quantifying uncertainty in medical decisions," *Journal of the American College of Cardiology*, vol. 14, no. 3, pp. A23–A28, 1989.

[31] J. Oden, T. Belytschko, J. Fish, T. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, and S. Yip, "Simulation-based engineering science: Revolutionizing engineering science through simulation–report of the national science foundation blue ribbon panel on simulation-based engineering science, february 2006."

[32] "Frederick Jelinek Memorial Workshops on Speech, Language and Computer Vision," 2016. [Online]. Available: http://www.clsp.jhu.edu/workshops/16-workshop/frederick-jelinek-memorial-workshops-on-speech-language-and-computer-vision/

[33] A. Hanbury, H. Müller, K. Balog, T. Brodt, G. V. Cormack, I. Eggel, T. Gollub, F. Hopfgartner, J. Kalpathy-Cramer, N. Kando, A. Krithara, J. Lin, S. Mercer, and M. Potthast, "Evaluation-as-a-Service: Overview and Outlook," *arXiv:1512.07454 [cs]*, Dec. 2015, arXiv: 1512.07454. [Online]. Available: http://arxiv.org/abs/1512.07454

[34] N. Habash, B. Dorr, and D. Traum, "Hybrid Natural Language Generation from Lexical Conceptual Structures," *Machine Translation*, vol. 18, pp. 81–127, 2003. [Online]. Available: http://ict.usc.edu/pubs/Hybrid%20Natural%20Language%20Generation%20from%20Lexical%20%20Conceptual%20Structures.pdf

[35] M. Johnson, J. M. Bradshaw, P. J. Feltovich, C. M. Jonker, B. van Riemsdijk, and M. Sierhuis, "The fundamental principle of coactive design: Interdependence must shape autonomy," *Lecture Notes in Computer Science: Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, vol. 6541, pp. 172–191, 2010.

[36] J. Lyons, K. K. Paliwal, A. Dehzangi, R. Heffernan, T. Tsunoda, and A. Sharma, "Protein fold recognition using hmm-hmm alignment and dynamic programming," *Theoretical Biology*, vol. 393, pp. 67–74, 2016.

[37] "NIST IAD Data Science Evaluation Workshop," Mar. 2016. [Online]. Available: http://www.nist.gov/itl/iad/mig/dseworkshop.cfm

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

An HL7 v2 Platform for Standards Development and Testing

S. Martinez[1] **and R. Snelick**[1]
[1]National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA

Abstract – *Development of healthcare data exchange standards has long been problematic, plagued with ambiguous and inconsistent requirement specifications. This situation leads to potential misinterpretation by implementers, thus limiting the effectiveness of the standard and creating artificial and unnecessary barriers to interoperability. Likewise, the ability to test implementations effectively for conformance to the standards is hindered. The current approach of standards development and test plan creation relies on word processing tools, meaning implementers must read and interpret the information in these documents and then translate it into machine-processable requirements and test assertions. This approach is error prone—a better methodology is needed. We present a set of productivity tools in an integrated platform that allow users to define standards and test plans that result in machine-processable artifacts. A testing infrastructure and framework subsequently uses these artifacts to create conformance testing tools automatically. We present and demonstrate the utility of a platform for developing standards, writing test plans, and creating testing tools. This end-to-end methodology is illustrated by describing a case study for the HL7 v2.6 Vital Records Death Reporting Implementation Guide.*

Keywords: Healthcare Data Exchange Standards; Healthcare Information Systems; Interoperability; Standards Development; Testing.

1 Introduction

For 30 years, HL7 Version 2 (v2) has been the predominant standard used for the exchange of healthcare administrative and clinical data. Healthcare information systems use the HL7 v2 protocol to develop standardized interfaces to connect to and exchange data with other systems. HL7 covers a broad spectrum of domains including Patient Administration, Laboratory Orders and Results, and Public Health Reporting. The base HL7 v2 standard [1] is a framework that contains many message events, and for each event it provides an initial template (starting point) that is intended to be constrained for a specific use case. The application of constraints to a message event is referred to as profiling [2,3]. For example, the ADT (Admit, Discharge, Transfer) A04 (Register a Patient) message event is a generic template for communicating information about a patient. The base message template is composed of mostly optional data elements. For a given use case, e.g., Vital Records Death Reporting (VRDR) [4], the message template is "profiled". That is, elements can be constrained to be required, content can be bound to a set of pre-coordinated codes, and so on. The base message event (e.g., ADT A04) that has been constrained for a particular use (e.g., VRDR) is referred to as a conformance profile. An implementation guide is a collection of conformance profiles organized for a particular workflow (e.g., report, revise, or cancel a death report). In this example, three conformance profiles exist each with different message events, one for report, revise, and cancel. To date, HL7 v2 implementation guides have been created using word processing programs, which has resulted in ambiguous and inconsistent specification of requirements. This practice has hindered consistent interpretation among implementers, which has created an unnecessary barrier to interoperability.

We present an end-to-end methodology and platform for developing standards (implementation guides), writing test plans, and creating testing tools in the HL7 v2 technology space [3,5]. The platform includes three key foundational components:

- A tool to create implementation guides and conformance profiles
- A tool to create test plans, test cases, and associated test data
- A testing infrastructure and test framework to build testing tools

A key to the approach is that the "normal" process of creating implementation guides, test plans, and testing tools is "reversed". Instead of creating requirements using a natural language and subsequently interpreting the requirements to create test plans and test assertions, the requirements are captured with tools that internalize the requirements as computable artifacts.

Figure 1 illustrates the methodology. Domain experts develop use cases, determine the message events that correspond to the interactions in the use cases, and then proceed to define the requirements. Using the NIST

Martinez, Sandra; Snelick, Robert.
"An HL7 v2 Platform for Standards Development and Testing."
Paper presented at HIMS'17 - The 3rd International Conference on Health Informatics and Medical Systems,
Las Vegas, NV. July 17, 2017 - July 20, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

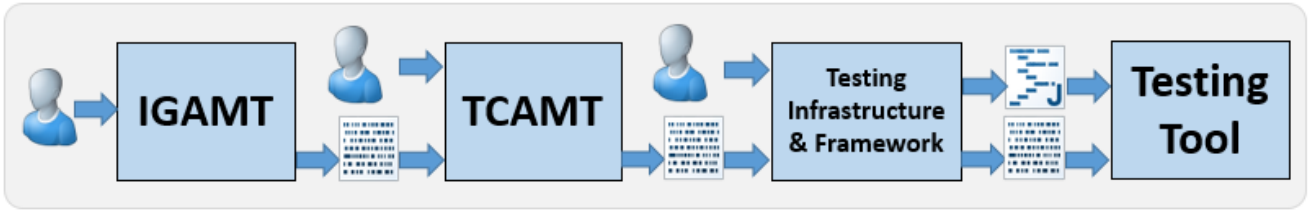


Fig. 1. NIST HL7 v2 Standards Development and Testing Platform

methodology, they accomplish these tasks by entering this information into the Implementation Guide Authoring and Management Tool (IGAMT). During this process, the domain experts constrain the message events according to the requirements needed by the use case. Section 2 will elaborate more on this process and on the details of how the requirements are constrained. The output of IGAMT is a set of artifacts that are represented in Word, HTML, and XML formats. The complete implementation guide, including the narrative and messaging requirements, can be exported in Word or HTML. Such formats are suitable for ballot at standards development organizations such as HL7 or IHE (Integrating the Healthcare Enterprises). Each conformance profile can be exported as XML. The XML format contains all of the messaging requirements in a machine processable representation, which is the most important aspect of IGAMT since the XML conformance profiles have many uses including message validation, test case and message generation, and source code generation.

The XML conformance profile and/or the internal IGAMT model are imported by the Test Case Authoring and Management Tool (TCAMT). TCAMT is used to create targeted test cases for interactions (profiles) defined in the implementation guide. The output is an additional set of constraints in an XML format. The entirety of the output generated from IGAMT and TCAMT is called a "resource bundle".

The NIST platform includes a testing infrastructure of common utilities used for testing, such as a message validation engine, along with a testing framework that provides various testing tool components, such as a communication framework and a profile viewer. Testing Tool instances are then created using the testing infrastructure and framework components as well as the resource bundle output generated from IGAMT and TCAMT.

The NIST platform in essence allows end users to create conformance testing tools by means of a set of productivity tools. This streamlined approach can greatly reduce today's problems with conformance test tools for standards. These problems include: there are too few of these tools, they are expensive to build, they are not dynamic for local refinements, and their time to market is protracted. Additionally, the platform provides value through enforcing consistent and rigorous rules for requirements specifications.

The remainder of this paper explains the NIST platform in more detail in the context of a real-world case study. The layout of the VRDR implementation guide is presented, and we describe how IGAMT is used to capture the messaging requirements. Next, an explanation of how a set of targeted test cases are created in TCAMT is provided. Finally, the resulting VRDR test tool is presented. The goal is to inform the reader about the ease with which HL7 v2 implementation guides, test cases, and testing tools can be created using the NIST platform compared to the current laborious methods used today.

2 IGAMT

IGAMT is a tool used to create HL7 v2.x implementation guides that contain one or more conformance profiles. The tool provides capabilities to create both narrative text (akin to a word processing program) and messaging requirements in a structured environment. Our focus in this paper is on the messaging requirements.

IGAMT contains a model of all the message events for every version of the HL7 v2 standard. Users begin by selecting the version of the HL7 v2 standard and the message events they want to include and refine in their implementation guide. For example, the message events ADT^A04, ADT^A08, and ADT^A11 are used to create 14 conformance profiles in the VRDR implementation guide. Each message event is profiled (constrained) to satisfy the requirements of the use case.

Rules for building an abstract message definition are specified in the HL7 message framework, which is hierarchical in nature and consists of building blocks generically called elements [1]. These elements are segment groups, segments, fields, components, and sub-components. The requirements for a message are defined by the message definition and the constraints placed on each data element. The constraint mechanisms are defined by the HL7 conformance constructs, which include usage, cardinality, value set, length, and data

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

type. Additionally, explicit conformance statements are used to specify other requirements that can't be addressed by the conformance constructs. The process of placing additional constraints on a message definition is called profiling. The resulting constrained message definition is called a conformance profile (also referred to as a message profile). An example of a constraint is changing ***optional*** usage for a data element in the original base standard message definition to ***required*** usage in the conformance profile.

IGAMT provides, in a table format user interface, the mechanisms to constrain each data element at each level in the structure definition. The rows of the table list the data elements according to the structure being constrained (segments, fields, and data types). The columns list the conformance constructs that can be constrained for a data element, including the binding to a value set.

One key philosophy of IGAMT is the capability of creating reusable building blocks. These lower level building blocks can be used to efficiently create higher level constructs. The building blocks include data type flavors, segment flavors, and profile components. A base data type can be constrained for a particular use; the resulting data type is called a data flavor (or data type specialization). A given base data type may have multiple data type flavors. These flavors can be saved in libraries and reused as needed. A similar process applies to creating segment flavors.

A profile component represents a subset of requirements that can be combined with other profiling building blocks. One such example is the definition of a profile for submitting immunizations. The CDC creates a national level profile. However, individual states may have additional local requirements that can be documented in a profile component. Only the delta between the national and local requirements is documented in the profile component. Combining the national level profile and the state profile component yields a *complete* (implementable) profile definition for that particular state. This design provides a powerful and effective approach to leveraging an existing profile.

A utility for creating and managing value sets is also provided. Specific value sets can be created and bound to data elements. Value set libraries can also be developed for reuse.

3 VRDR Use Cases

The Vital Records Death Reporting (VRDR) Implementation Guide (IG) [4] was developed to support the transmission of death-related information from the health care provider's electronic health records (EHR) to the jurisdictional vital records offices (JVRO) and to the National Statistical Agency (NSA) [6]. Five use cases/workflows are identified to describe the transmission of data: Provider Supplied Death Information (PSDI), Jurisdiction Death Information (JDI), Void Certificate Reporting (RVCA), Coded Cause of Death (CCODA), and Coded Race/Ethnicity (CREIA) [4]. The use cases require three message events: ADT^A04, ADT^A08 and ADT^A11. A given use case has more than one interaction; in total, 14 interactions are needed.

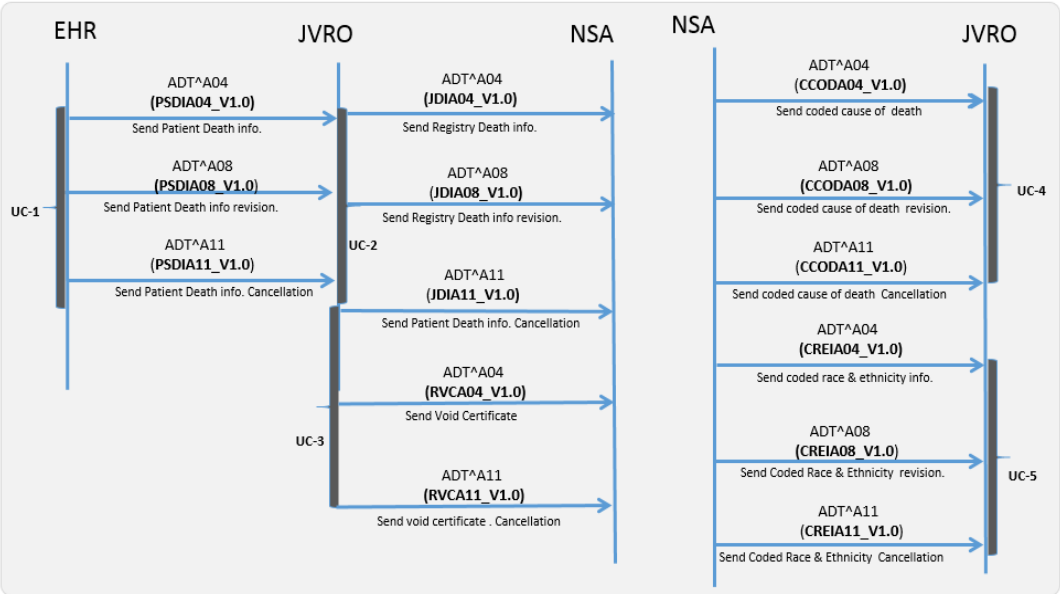


Fig. 2. Vital Records Death Reporting Interactions

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

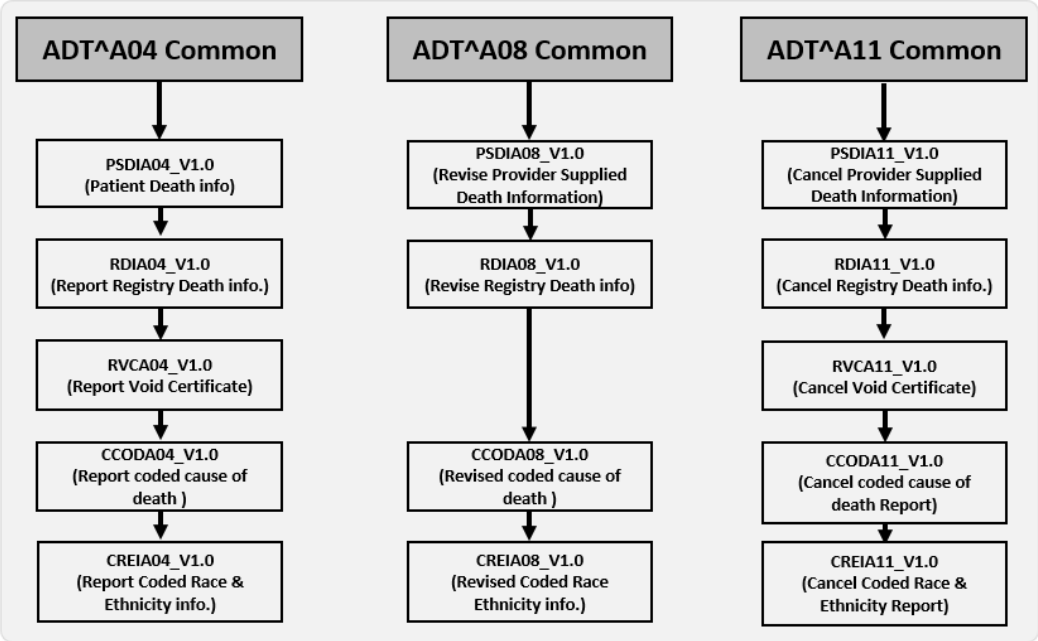


Fig. 3. VRDR Profiles and Design

Figure 2 shows the 14 interactions supporting the VRDR use cases. The type of information being exchanged determines how each interaction (message) is constrained. For the PSDI use case, the ADT^A04 is constrained for reporting about a person's death, the ADT^A08 message is constrained for updates to the report, and the ADT^A11 message is constrained to cancel the report.

Each interaction is assigned a unique profile identifier; e.g., "PSDIA04_V1.0" is a profile identifier for the "Send Patient Death Information" interaction (ADT^A04 interaction in Figure 2). The same three message events (ADT^A04) are employed across various use cases, however, the context in which they are used is different; therefore, the set of constraints applied are different, each resulting in a unique conformance profile (for the same base ADT^A04 event). The content is defined by the set of initiating and responding systems.

Figure 3 shows the conformance profile-building approach for the VRDR profiles and the group of profiles that share the same trigger event. The A04 message event is loaded and constrained to create the common A04 profile. From there, the PSDIA04_V1.0 profile is created. The PSDIA04_V1.0 profile is used in building all of the profiles associated with the A04 event, and the corresponding message-level constraints are added to each profile. Constraints at the message level include segment and group usage, cardinality, and any additional requirement in the form of conformance statements.

The approach used in the development of the A04 profiles is followed in the creation of the A08 and A11 profiles. The base message events are loaded and constrained to develop the PSDIA11_V1.0 and PSDIA08_V1.0 profiles. These profiles are then leveraged to create the remaining profiles sharing the same message event using the IGAMT cloning capability.

Profiling at the value set, segment, field, and data type (component) levels is followed, and it can be achieved in any order, thus taking advantage of the IGAMT capability that allows for the creation of reusable building blocks. The value set library can be created using the IGAMT built-in mechanism for loading value sets from HL7 tables and the CDC PHINVADS (Public Health Information Network Vocabulary Access and Distribution System) value sets [7].

VRDR data type flavors are built from the HL7 v2.6 data type library. They are defined using the constraints specified in the IG, such as length, usage, and value set binding and any constrains in the form of conformance statements. Occasionally, depending on how the data type is going to be used, the IG defines more than one data type specialization for a base data type. In the case of the "HD (Hierarchic Designator Assigning Authority)" data type, an additional flavor called HD_AA is defined. This flavor is used when an OID (Object Identifier) is assigned to designate an assigning authority.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The VRDR segment flavor are created from the HL7 v2.6 segment library using the length, data type, cardinality, usage, value set binding, and conformance statement constraints defined in the IG. In some instances, it is necessary to create additional segment flavors to indicate constraint deltas among the profiles. For example, the constraints in the PV1 segment are applied to every profile, therefore only one PV1 flavor is created. In the case of the PID segment, the constraints are different in each use case; therefore, a segment flavor is created for each use case, and that flavor can be reused in the respective profiles.

As shown, a key feature in IGAMT is the capability to create precise object definitions and to use (and reuse) the objects as building blocks to create higher level objects, such as segment and conformance profiles.

4 TCAMT

TCAMT is a tool used to create HL7 v2.x test plans that contain one or more (typically many) test cases. A test case can consist of one or more test steps. A test step can be an HL7 v2.x interaction or a manual step such as visually inspecting the contents of the system under test's (SUT's) display screen. Each test case and test step can consist of a test description, pre- and post-conditions, objectives, evaluation criteria, and additional notes and comments. Test steps for an HL7 v2.x interaction contain an HL7 v2 message (that is, specific data) that is in alignment to an XML conformance profile created from IGAMT.

TCAMT allows domain experts to create test cases that target certain scenarios and capabilities. Using these test cases provides context, which expands the scope of testing beyond just the constraints in the conformance profile. Without context, a validation tool cannot test a message exhaustively to all requirements specified in the implementation guide. For example, elements with "required, but may be empty (RE)" usage or elements with "conditional usage (C)" cannot be assessed without targeted tests. A message that is validated against the requirements of a conformance profile without any provided context is called "context-free testing". A message that is validated against the requirements of a conformance profile and with a provided context is called "context-based testing" [3]. The test cases provide context, and TCAMT is a tool that allows users to create the test cases.

A key feature in TCAMT is its use of the conformance profiles created in IGAMT as a foundation. The message definition and requirements are available to the TCAMT user based on information that was entered into IGAMT. Then, the TCAMT user provides the data associated with each message element of interest. TCAMT also allows the user to enter additional assertion indicators based on what they want to test. For example, for an element with a usage of "RE", the user might provide data that are expected to be entered into the sending system for the element, and the user also might select an assertion indicator. There are several assertion indicators that could be selected, for example, "presence". In this case, if the user provides test data and the indicator of "presence", an additional assertion (constraint) is generated by TCAMT and is provided to the validation. For elements with "RE" usage, the element must be supported by the SUT, but in a given message instance the element may not be populated. For this construct, the tester wants to ensure that the implementation has, in fact, included support for the element.

In a context-free environment, the absence of data in a message is not a conformance violation for elements with "RE" usage. However, in the example test case described above, data were provided, and an assertion for the presence of the data was selected. Now, when a message created for this test case is validated, the additional assertion triggers the check for the presence of data for this element. This method is one way to determine support for the element.

Via TCAMT, the user can create an unlimited number of test cases and test a broad spectrum of requirements. Other assertion indicators can be used to test for specific content or for the non-presence of an element. Additionally, test data can be provided to trigger conditional elements. In other instances, support for certain observations may need to be ascertained. In such cases, test data for specific observations (e.g., cause of death, date/time pronounced dead, etc.) are provided, requiring the message instance to contain an OBX segment for that observation. TCAMT provides the mechanisms to conveniently and consistently create test cases. Output from TCAMT provides the additional constraints that are interpreted by the validation engine.

5 VRDR Test Plan

The VRDR test plan consists of a set of scenarios and test cases that emulate real-world events and workflow. The scenarios are designed to target specific requirements that are not easily testable in a context-free environment. The goal of the VRDR test plan is to provide a set of test cases that collectively tests the spectrum of requirements defined in the VRDR implementation guide. Therefore, for each interaction in support of a use case, test scenarios and test cases are needed.

Figure 4 shows an excerpt of the test plan. A scenario for the Provider Supplied Death Information (PSDI) is indicated that contains three test cases and associated test

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

steps. Test steps have a 1-to-1 relationship to an HL7 v2 message (interaction), and each message is bound to the requirements in its corresponding conformance profile.

Fig. 4. VRDR Test Plan Excerpt

For each test case, a real-world story is given along with specific test data that coincide with the test story. The test data provide a known data set that can be used to create additional assertions (beyond those provided in the conformance profile). This approach is the principle behind context-based testing. Each test step interaction contains a message that is associated with its corresponding conformance profile. TCAMT provides a productivity mechanism to create the test messages using the underlying structure provided in the conformance profile. Once the test message (and therefore test data) has been created, additional assertions can be specified that align with the testing goals.

Table 1 shows two important examples of how this process works. TCAMT facilitates specific assertions by allowing the test plan designer to assign assessment indicators to the data elements. The combination of the provided test data and assessment indicator generates the assertions. The PDA-2.9 example shows a case where the Death Location Description element is constrained with "RE" usage, which indicates that the element must be supported but data may not always be available. To test support for this element, the test case provides test data ("Mercy Hospital") and the assessment indicator is set to "Presence". These settings will generate an assertion that makes the presence of the PDA-2.9 element required. The OBX-3.1 example shows a case where an observation (an OBX segment) is expected in which the observation is "Cause of Death". Here, the value "69453-9" is provided, which is a LOINC (Logical Observation Identifiers Names and Codes) code that indicates a "Cause of Death". By explicitly requiring the "Cause of Death" observation be included in the message, the testing is ensuring that the SUT can support this observation. In this example, only one element in this segment is shown, but typical testing scenarios will have a coordinated set of assertions for the set of elements in the OBX segment. For example, OBX-3.3 would assert that the content of this element is "LN" to confirm that the code "69453-9" is in fact drawn from the LOINC code system.

Creating test cases that target specific capabilities, such as "sending the Cause of Death observation" is an important aspect of testing and a key incentive for conducting context-based testing. Without this level of specificity, assessment of systems is limited. Only a few examples have been provided here to give the reader a sense of the sorts of items that can be tested. However, this approach expands the test space significantly. Other aspects that can be tested include cardinality, length, value set constraints, conditional elements, specific content, workflow, and functional requirements.

Element	Name	Usage	Test Data	Assessment Indicator	Rationale	Conformity Assessment
PDA-2.9	Death Location Description	RE	Mercy Hospital	Presence	Assess that the SUT can support this element.	PDA-2.9 SHALL be present.
OBX-3.1	Observation Identifier	RE	69453-9	Value-Test Case Fixed	Assess support for providing the "Cause of Death" observation	An OBX segment SHALL be present in which OBX-3.1 SHALL be "69453-9".

Table 1. Context-based Assertion Examples

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

6 Infrastructure and Framework

NIST has built an HL7 v2.x testing infrastructure and framework to aid in the process of creating conformance testing tools. The testing infrastructure provides a set of services utilized by the test tool framework to build specific instances of tools. A test tool can be specific for a particular domain, or it can be general-purpose. The general-purpose tool is a NIST-hosted web application where a user can upload conformance profiles and test plans to create a test tool. The conformance test tool essentially is generated as a by-product "for free" once the validation artifacts have been created. This liberates the domain experts from the tool building process. Alternatively, the framework can be leveraged, customized, and installed locally. Using the framework, developers can choose to create domain specific or general-purpose web application conformance test tools, access the validation via web services, or incorporate validation via JAR (Java Archive) files or source code. Regardless of the use, the NIST platform can significantly improve the quality of implementation guides, assist in the creation and maintenance of test plans, expedite the stand-up of a validation tool, and, overall, reduce the cost and time of the entire process.

7 VRDR Test Tool

A VRDR conformance testing tool is built using the testing infrastructure and framework, the IGAMT-produced conformance profiles, and the TCAMT-produced test plan. The test tool is a web-based application (see [8] to access) that supports both context-free and context-based validation. In addition to performing message validation, the tool provides a browse-able view of the requirements for each conformance profile. In the context-based mode, the test story, test data, and an example message are provided for each test step.

In the context-free mode, the user simply selects the conformance profile to validate against and imports the message. The validation is performed automatically and a report is given. In the context-based mode, the user selects the test step and imports the message to validate. The test tool sets the validation to the conformance profile linked to the test step, performs the validation, and provides a report. In both modes, a tree structure of the message is shown on the left panel of the validation screen and can be used to inspect the content of individual data elements.

8 Summary

We presented an end-to-end methodology and platform for developing standards, writing test plans, and creating testing tools in the HL7 v2 technology space. The platform includes three key foundational components: (1) a tool to create implementation guides and conformance profiles; (2) a tool to create test plans, test cases, and associated test data; and (3) a testing infrastructure and test framework to build testing tools. We demonstrated the approach by creating a test tool for the HL7 v2.6 Vital Records Death Reporting use case. Requirements were captured in IGAMT and exported as conformance profiles. TCAMT was used to create a set of test cases based on the conformance profiles. A conformance test tool was created by combining the validation artifacts with the testing infrastructure and framework.

9 References

[1] Health Level 7 (HL7) Standard Version 2.6, ANSI/HL7, October 2007, http://www.hl7.org.

[2] *Principles for Profiling Healthcare Data Communication Standards*. R. Snelick, F. Oemig. 2013 Software Engineering Research and Practice (SERP13), WORLDCOMP'13 July 22-25, 2013, Las Vegas, NV.

[3] *Healthcare Interoperability Standards Compliance Handbook*. F. Oemig, R. Snelick. Springer International Publishing Switzerland, ISBN 978-3-319-44837-4, December 2016.

[4] *HL7 Version 2.6 Implementation Guide: Vital Records Death Reporting, Release 1*. Draft Standard for Trail Use. August 2016. http://www.hl7.org.

[5] *NIST Resources and Tools in Support of HL7 v2 Standards*. http://hl7v2tools.nist.gov/

[6] *CDC National Vital Statistics System: http://www.cdc.gov/nchs/nvss/evital_standards_intiatives.htm*

[7] CDC Public Health Information Network Vocabulary Access and Distribution System (PHIN VADS); https://phinvads.cdc.gov/.

[8] *NIST Vital Records Death Reporting (VRDR) Conformance Testing Tool*; http://hl7v2-vr-r2-testing.nist.gov

Imposing Fine-grain Next Generation Access Control over Database Queries

David Ferraiolo, Serban Gavrila, Gopi Katwala, and Joshua Roberts

National Institute of Standards and Technology
Gaithersburg, Maryland 20899
{dferraiolo, Serban.gavrila, gopi.katwala, joshua.roberts}@nist.gov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

ABSTRACT

In this paper, we describe a system that leverages ANSI/INCITS Next Generation Access Control (NGAC) standard called Next-generation Database Access Control (NDAC) for accessing data in tables, rows, and columns in existing RDBMS products. NDAC imposes access control at the data level, eliminating the need for implementing and managing access control in applications and/or through the use of proprietary RDBMS mechanisms. Consequently, the same policies can protect multiple databases from queries sent from multiple applications. Furthermore, NDAC not only provides control down to the field level, but to varying fields of select rows. NDAC is unique in achieving this granularity of control without the use and coordination of multiple ptotection mechanisms. Operationally, users issue wide sweeping queries, and NDAC allows access to the optimal amount of data permissible for the user. The method includes an Access Manager for trapping and enforcing policy over SQL queries issued by applications as well as a Translator for converting SQL statements to NGAC inputs and converting NGAC authorization responses to either an access Deny or one or more permitted SQL statements.

Keywords

ABAC; NGAC; Policy Machine; DBMS; Access Control

1. INTRODUCTION

Relational Database Management Systems (RDBMSs) do not typically impose access control directly on its data. To restrict access to sensitive data that might reside in a RDBMS, controls are typically implemented at the application level or through propriety RDBMS methods such as views. These controls take on many forms including role-based access to "screens" with parameters that can be characterized and subsequently used to formulate and issue SQL queries. SQL queries comprise four basic types of operations–Select, Insert, Update, and Delete–that respectively read, create, write, and delete data in tables. An important feature of RDBMSs is that they are able to specify criteria and extract and/or alter data that might reside in one or more tables with great efficiency. For example, "give me all the employees over 50 years old that live in Virginia".

This paper is authored by an employee(s) of the United States Government and is in the public domain. Non-exclusive copying or redistribution is allowed, provided that the article citation is given and the authors and agency are clearly identified as its source.
ABAC'17, March 24 2017, Scottsdale, AZ, USA
ACM 978-1-4503-4910-9/17/03
DOI: http://dx.doi.org/10.1145/3041048.3041050

In this paper we describe a method that leverages ANSI/INCITS Next Generation Access Control (NGAC) standard [1] [2] called Next-generation Database Access Control (NDAC) for imposing access control over database queries at the data level, independent of the application and with minimal impact on performance. As a result, the same policies can protect multiple databases from queries sent from multiple applications.

NDAC's method of protection begins with automatically generated composite objects in the form of object attributes from a database schema and the expression of access control policies in terms of those attributes. NDAC uses NGAC as an authorization engine to manage access control policies (through its Policy Administration Point (PAP)) and compute authorization responses (through its Policy Decision Point (PDP)). [3] provides an open source for such an engine. The method also includes an Access Manager (a customized NGAC Policy Enforcement Point (PEP)) for trapping and enforcing policy over SQL queries issued by applications and a Translator for converting SQL statements to NGAC inputs and converting NGAC authorization responses to either an access Deny or one or more permitted SQL statements.

Furthermore, NDAC provides control down to the granularity of select rows with varying fields. Operationally, users issue wide sweeping queries, and NDAC allows access to an optimal set of permissible data. Although other technologies (see section 6–Related Work) achieve a similar granularity of protection through the combined use of multiple protection mechanisms, NDAC is unique in its use of just one policy store. The principle advantage is that NDAC does not need to maintain and coordinate multiple access control schemes and can use the same policy store to protect non-RDBMS resources, such as files, using an NGAC standards PEP.

To demonstrate viability and assess performance, we have created an NDAC prototype/experimental implementation using Harmonia 1.6–an NGAC reference implementation that uses MySQL for its access control database [3]. For purposes of computing a decision or reviewing access rights, all information that is needed resides in memory. Harmonia 1.6 access control information is loaded from disk into memory when the PDP is initialized and updated when an administrative change occurs.

The remainder of this paper focuses on the method rather than the NDAC prototype/experimental implementation due to its early stage of development.

2. NGAC OVERVIEW

The Policy Machine (PM) [4] is an access control framework that

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

served as the basis for the development of an ANSI/INCITS standard call Next Generation Access Control (NGAC). NGAC consists of:

- a standard set of data elements and relations that can be configured to express arbitrary access control policies in support of a wide variety of data services and applications
- a generic set of operations that include read/write operations that can be performed on resource data as well as administrative operations for configuring (creating and deleting) the data elements and relations that represent policies
- a standard set of functions for computing access control decisions and enforcing policy over user access requests to perform read/write and its administrative operations

NGAC is a flexible access control framework in that it can be molded in support of multiple combinations of diverse access control policies. NGAC can often provide much of the same data service functionality that is supplied by existing application products and system utilities (e.g., file management, workflow, internal messaging) and with similar performance [5]. An advantage of NGAC is that access control policies are comprehensively enforced over its data services, while non-NGAC data service counterparts lack such faculties. Although it is possible to develop a NGAC relational DBMS data service with features similar to today's commercially available RDBMS products, the NGAC data service performance would pale in comparison. Furthermore, the NGAC-enabled RDBMS data service could not directly accommodate the broadly recognized SQL standard for accessing databases.

3. NDAC

NDAC provides a means of leveraging NGAC for expression and enforcement of access control policies over SQL queries for accessing data in tables, rows, and columns in existing RDBMS products. By leveraging NGAC, the method provides a means of access control policy support that goes beyond state-of-the-art with minimal impact on performance. It can impose forms of mandatory, discretionary, and history-based access control policies [6]. Architecturally, NDAC could be deployed externally to RDBMS, thereby providing a general solution for a variety of RDBMS products, or it could be implemented as a database-kernel module.

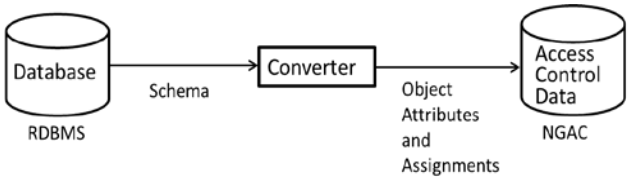


Figure 1. Converting Database Schema to NGAC Access Control Data

Included among NGAC's data elements and relations used to express and enforce policies are Object Attributes. Object Attributes are containers that group and characterize data objects in diverse ways. Data objects and object attributes are placed into containers through an assignment relation. Vis. Figure 1, the NDAC process for expressing access control policies begins with an existing RDBMS schema, which includes columns and tables that are automatically converted into NGAC-corresponding object attributes and assignments.

Given that rows are also object containers, existing rows could be automatically converted as well. NGAC data elements and relations also include User Attributes, a generic set of operations, and three types of relations for specifying an access policy. Once the RDBMS schema has been converted, NGAC relations are configured in formulating policy in terms of the created object attributes and assignments using NGAC's administrative API. The resulting data elements and relations are stored as NGAC Access Control Data. In addition to the conversion and supplementary data elements and relations, NDAC includes an Access Manager for trapping SQL queries from applications, a Translator for converting SQL queries along with a user identity to NGAC inputs, and NGAC authorization responses to those inputs to either an access deny or permitted SQL queries.

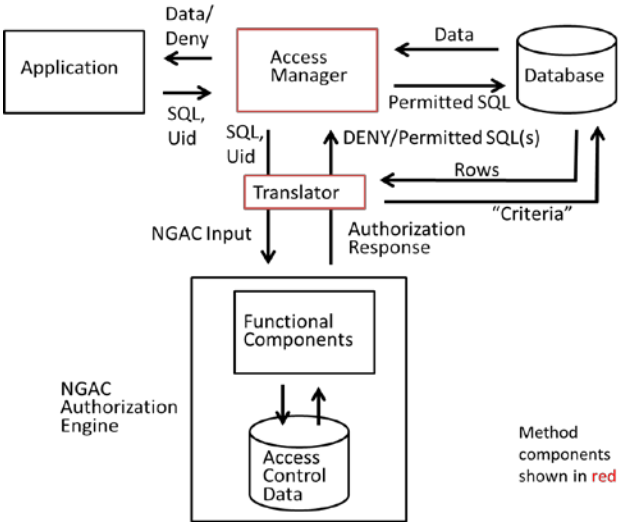


Figure 2. Placement of NDAC with respect to existing components

Figure 2 shows the placement of NDAC's Access Manager and Translator in an authorization flow that involves Applications, a target Database, and an NGAC authorization Engine. The authorization flow is as follows:

(1) The SQL statement from a user of the Application is intercepted by the Access Manager and sent to the Translator.
(2) For Select, Update, and Delete statements, using a separate transaction, identify the set of rows that meet the criteria included in the SQL statement. For Insert, this step is not used.
(3) The Translator converts the SQL statement from the user into NGAC inputs that are fed to an NGAC implementation (engine).
(4) Using its Access Control Data, and the rows identified in (2), the NGAC implementation computes and renders an Authorization Response that is sent back to the Translator.
(5) The Translator converts the Authorized Response into either an access DENY or one or more SQL Statements that are permitted for the user and sent back to the Access Manager.
(6) The Access Manager submits the Permitted SQL Statements to the Database.
(7) In the case of a Select operation, Data extracted from the database is sent back to the Access Manager and forwarded to the Application and user.

Depending on the type of query (Select, Update, Insert, or Delete) the Translator issues different inputs to the NGAC Authorization

Engine. These details are discussed later in the paper.

4. EXPRESSING POLICIES

4.1 Basic Elements, Containers, and Relations

NGAC access control data includes users, data objects, generic operations, and user and object attributes among its elements. NGAC treats both user attributes and object attributes as containers. Containers are instrumental in both formulating and administering access policies and attributes. NGAC expresses access policies through configurations of relations that include assignments (define membership in containers), associations (to derive privileges), and prohibitions (exceptions to privileges).

User attribute containers characterize their members. These containers can represent user names, roles, affiliations, or other common characteristics pertinent to policy such as security clearances.

Object attribute containers characterize data by identifying collections of objects such as those associated with certain projects, applications, or security classifications. Object containers can also represent tables, columns, and rows.

NGAC uses a tuple (x, y) to specify the assignment of element x to element y. The assignment relation always implies containment (i.e., x is contained in y).

Users and objects may be contained in one or more containers, and containers may be contained by or contain other containers of the same type. For object containers, this allows for the representation of complex data structures such as relational database tables with distinguished fields. Rows of a table may be expressed as containers of data objects corresponding to the row's fields, and columns may be expressed as containers of data objects corresponding to column fields. Figure 3(b) illustrates a table using ovals to represent containers and dots to represent individual data objects. The vertically oriented ovals represent columns (Name, Phone, SSN, and Salary), the horizontally oriented ovals represent rows (AliceRecord, BobRecord, and TomRecord), and their intersections represent fields in one or more tables. Figure 3(b) further illustrates a container of rows (Gr2Records) and two containers of columns (Public and Sensitive). All rows and all columns are represented by the object container EmployeeTable.

Note that for this example, the containers shown in red are the object attributes that were automatically created by the Converter (see figure 1). All other NGAC elements and relations are assumed to be created through an NGAC administrative API by an authorized user. This authorized user may be a policy administrator or, as we discuss later, the user submitting Insert or Delete SQL queries.

Figure 3(a) illustrates user containers (also called user attributes) for the grouping and characterization of users. The container named Staff includes three users (u1, u2, and u4), and the container HR includes two users (u3, and u5). Employee is a container of containers (HR and Staff). In addition, figure 3(a) shows three containers—Bob, Alice and Tom—that respectively contain u1, u2, and u4. Finally, figure 3(a) shows Gr2Mng containing user u2.

NGAC recognizes a generic set of operations that include basic input and output operations (i.e., read and write) that can be performed on the contents of data objects as well as a standard set of administrative operations that can be performed on NGAC data elements and relations that represent policies and attributes.

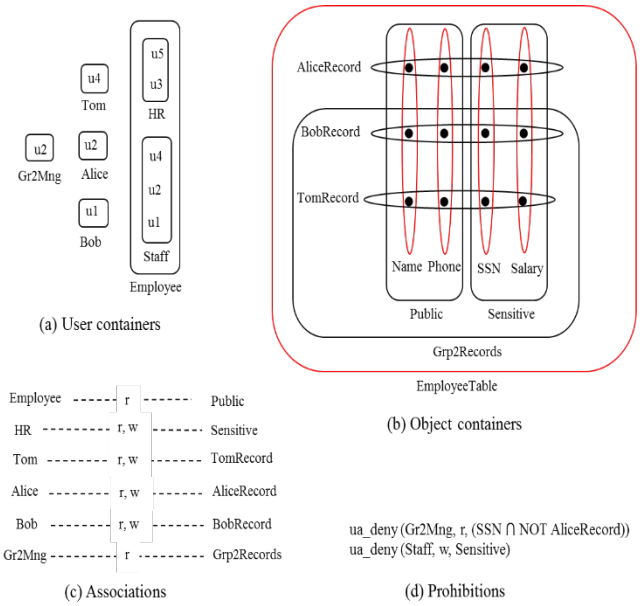


Figure 3. Example Policy Configuration

To carry out an operation, one or more access rights are required. As with operations, two types of access rights apply: non-administrative access rights and administrative access rights.

4.2 Associations

Access rights to perform operations are acquired through associations. An association is a triple, denoted by *ua---ars---pe*, where *ua* is a user attribute, *ars* is a set of access rights, and *pe* is a policy element that may comprise either a user attribute or an object attribute. The policy element *pe* in an association is used as a reference for itself and the policy elements contained by the policy element. The meaning of the association *ua---ars---pe* is that the users contained in *ua* can execute the operations enabled by the access rights in *ars* on the policy elements referenced by *pe*. The set of referenced policy elements are dependent on (and meaningful to) the access rights in *ars*.

Figure 3(c) lists six association relations in terms of the user and object attributes (containers) illustrated in figures 3(a) and 3(b). The set of referenced policy elements are dependent on the access rights in *ars*. Note that the policy element of each association is an object attribute and the access rights are read/write. In the association HR---{r, w}---Sensitive, the policy elements referenced by Sensitive are data objects (dots) contained in Sensitive, meaning that user u3 and u5 can read and write those objects. If we had an association HR---{create assign-to}---Sensitive, where "create assign-to" is an administrative access right, then the policy elements referenced by Sensitive would be Sensitive, SSN, and Salary, meaning that users u3 and u5 may create assignments to Sensitive, SSN, or Salary.

The access policy specified by the list of associations in figure 3(c) is as follows:

- Employee users can read Name and Phone fields of all records in EmployeeTable

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

- In addition to being able to read Name and Phone fields, HR users can read and write SSN and Salary fields of all records in EmployeeTable
- Bob, Tom, and Alice can read and write all fields (SSN, Salary, Name, and Phone) in their own record (respectively, BobRecord, TomRecord, and AliceRecord)
- Gr2Mng can read all fields (SSN, Salary, Name, and Phone) of all records in Gr2Reccords (i.e., BobRecord and TomRecord)

4.3 Prohibitions

In addition to assignments and associations, NGAC includes three types of prohibition relations. In general, prohibition relations specify privilege exceptions. One of these relations is user attribute-deny. The user attribute-based deny relation is denoted by ua_deny(*ua*, *ars*, *pes*), where *ua* is a user attribute, *ars* is an access right set, and *pes* is a policy element set used as a reference for policy elements contained by the policy element(s). The meaning of the relation is that the users assigned to *ua* cannot execute the operations enabled by the access rights in *ars* on the policy elements in *pes*.

Figure 3(d) lists two prohibitions. The first prohibition specifies that users assigned to Gr2Mng cannot read objects in SSN with the exception of objects in AliceRecord. The second prohibition specifies that users assigned to Staff cannot write to objects in Sensitive.

The prohibitions listed in figure 3(d) further constrain the access policy as follows:

- Staff users can read Name and Phone fields of all records in EmployeeTable
- In addition to being able to read Name and Phone fields, HR users can read and write SSN and Salary fields of all records in EmployeeTable
- Bob, Tom, and Alice can read all fields (SSN, Salary, Name, and Phone) and write to Name and Phone fields in their own record (respectively, BobRecord, TomRecord, and AliceRecord)
- Gr2Mng can read all fields of all records in Gr2Records with the exception of the SSN field

An example set of Employee Records with data content is shown in the top table of figure 4 under the object containers depicted in figure 3(b). The bottom three tables show the access capabilities for users u1, u2, and u3 under the access control policy expressed in figure 3, where read access is highlighted in black, and read/write access is highlighted in red.

5. TRANSLATOR

As discussed in section 3, the NDAC includes a Translator. On one side, the Translator converts an SQL statement generated by an application and the identity of the application's user to an NGAC input. On the other side, the Translator takes an NGAC authorization response to the input and converts it to either one or more permitted SQL statements; an access DENY in the case of a Select statement; or to a GRANT or DENY status in the case of an Update, Insert, or Delete statement. The Translator treats Select and Update operations differently than Insert and Delete Operations since Select and Update operations are directly mapped to NGAC read and write operations on data. Alternatively, Insert and Delete operations are mapped to create and delete administrative operations on NGAC object containers that correspond to rows.

Some Employee Records

Name	Phone	SSN	Salary
Bob	301-976-4454	122-54-4537	$38,341
Alice	301-976-3042	945-39-4034	$72,440
Tom	301-976-2067	304-75-3995	$62,550

u1 (Bob, Staff)

Name	Phone	SSN	Salary
Bob	301-976-4454	122-54-4537	$38,341
Alice	301-976-3042		
Tom	301-976-2067		

u2 (Alice, Staff, Gr2Mng)

Name	Phone	SSN	Salary
Bob	301-976-4454		$38,341
Alice	301-976-3042	945-39-4034	$72,440
Tom	301-976-2067		$62,550

u3 (HR, Staff)

Name	Phone	SSN	Salary
Bob	301-976-4454	122-54-4537	$38,341
Alice	301-976-3042	945-39-4034	$72,440
Tom	301-976-2067	304-75-3995	$62,550

Figure 4. Example set of records with data content and the access capabilities for users u1, u2, and u3 under the access control policy of figure 3

5.1 Select and Update

Select SQL statements include a specification of one or more tables and one or more columns from those tables along with criteria for identifying rows from the table(s). Update SQL statements include a specification of one table with one or more columns with criteria for identifying rows. The method for translating a user's requested Select statement to one or more permitted SQL statements or an Update statement to a GRANT or DENY result is based on NGAC's ability to review the access capabilities of users. See [7] for a linear time algorithm and method for reviewing NGAC user capabilities. In particular, NDAC identifies a set of objects that are accessible to a user for either read for Select or write for Update as well as attributes that contain those objects. In the algorithms that follow, the terms "row," "column," and "table" refer to object attributes that correspond to those entities. Possible algorithms for Select and Update are as follows:

For Select:

(1) Using a separate transaction, identify the set of rows in the SQL database that meet the criteria included in the Select SQL statement.
(2) For each row identified in (1), identify a maximal set of columns that are a subset of the columns in the Select statement, and each identified column contains an object (for which the user has Read access) that is also contained in the row. These columns are said to be associated with the row.
(3) For each row, column association, remove the columns that are also included in any DENY relation for the user with respect to Read.
(4) For each subset of identified rows so that each row in the subset has a common associated set of columns, generate a Select SQL statement for that set of columns with the original table and original condition augmented by a condition that limits the Select to the subset of identified rows.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

(5) If the set of rows or columns are empty, the Translator issues a DENY response.

For Update:

(1) Identify the set of rows in the SQL database that meet the criteria included in the Update SQL statement.
(2) Identify a set of rows in the table of the Update SQL statement containing objects accessible by the user under the write operation.
(3) If the rows identified by (1) are a subset of those identified in (2), proceed to (4). Otherwise, DENY access.
(4) For each row identified in (1), verify the existence of objects common to the row and the set of columns included in the SQL Update statement. If the condition fails, DENY access. Otherwise, proceed to (5).
(5) For the columns included in the SQL Update statement, verify that the columns are not included in any deny relation for the user. If the condition holds, GRANT the SQL Update Statement. Otherwise, DENY access.

To provide a sense of potential performance, preliminary data shows that the NDAC prototype/experimental implementation currently computes and displays the results of authorizations of 100 records with 6 fields in 4 seconds and 1,000 records in 40.2 seconds.

5.2 Delete and Insert

The execution of an SQL Delete statement removes one or more rows from a table in accordance with criteria included in the statement. NDAC Grants or Denies a user's request to delete one or more rows in a database table, and, in the case of a Grant, subsequently deletes the corresponding NGAC object attributes and relations. The execution of a SQL Insert statement creates a new row with specified column values in a specified table. The method either Grants or Denies a user's request to insert a row in the database, and, in the case of a Grant, subsequently creates an NGAC object attribute corresponding to the row, creates objects (representing the values), and assigns those objects to the row attribute and appropriate column attributes. A user's capability to perform an SQL Delete or Insert operation is dependent on the existence of administrative privileges.

The creation and deletion of objects, object attributes, and assignments is achieved through the execution of administrative operations. A user's capabilities to execute administrative operations are established through administrative privileges.

5.2.1 Administrative Operations

Administrative operations in NGAC are implemented using parameterized routines, prefixed by a precondition, with a body that describes how a data set or relation (denoted by Y) changes to Y′. The precondition tests the validity of the actual parameters. If the condition evaluates to false, then the routine fails:

Rtnname (x_1, x_2, …, x_k) {
…*preconditions*…
{
Y′= f(Y, x_1, x_2, …, x_k)
}

Consider as an example the administrative operation CreateOinOA shown below, which specifies the creation of an object x and assigns the object to an object attribute y. The preconditions here stipulate that the x parameter is not a member of objects (O), and the y parameter is a member of object attributes (OA). The body describes the addition of the x to the set of objects (O), which changes the state of the set to O', and the addition of the tuple (x, y) to the set of assignments (ASSIGN) relation, which changes the state of the relation to ASSIGN'.

CreateOinOA(x, y)
x ∉ O ⋀ y ∈ OA
{
O' = O ⋃ {x}
ASSIGN' = ASSIGN ⋃ {(x, y)}
}

Each administrative routine entails a modification to the NGAC configuration.

5.2.2 Administrative Privileges

In order to execute an administrative operation, the requesting user must possess appropriate access rights. Just as access rights to perform read/write operations on data objects are defined in terms of associations, so too are capabilities to perform administrative operations on policy elements and relations.

For example, consider the following two associations in support of the configuration depicted by Figure 3(b):

TableAdmin---{*create-oa*, *create-o*, *create ooa*}---EmployeeTable

TableAdmin---{*delete-o*, *delete-oa*, *delete-ooa*, *delete-oaoa*}---EmployeeTable

The meaning of the first association is that a user assigned to TableAdmin can:

(1) create an object attribute (e.g., corresponding to a row) assigned to an object attribute (e.g., EmployeeTable) in EmployeeTable
(2) create an object assigned to an object attribute (e.g., an existing row) in EmployeeTable
(3) create an object to object-attribute assignment from an object (e.g., an object in a row) to an object attribute (e.g., corresponding to a column) in EmployeeTable

The meaning of the second association is that a user assigned to TableAdmin can:

(1) delete an object to object-attribute assignment (e.g., delete object assignments to attributes corresponding to a row and column) in EmployeeTable
(2) delete an object in EmployeeTable
(3) delete an object-attribute to object-attribute assignment (e.g., a row assigned to EmployeeTable) in EmployeeTable
(4) delete an object attribute (e.g., corresponding to a row) in EmployeeTable

5.2.3 Administrative Routines

The administrative operations necessary to insert or delete an object container corresponding row in another object container

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

corresponding to a table do not need to be executed on an individual basis, but instead can be executed as an NGAC administrative routine.

An *administrative routine* consists mainly of a parameterized interface and a sequence of administrative operation invocations. The body of an administrative routine is executed as an atomic transaction—an error or lack of user privileges that causes any of the constituent operations to fail execution subsequently causes the entire routine to fail, producing the same effect as though none of the operations were ever executed.

The following routine (in the context of figure 3(b)) creates an object attribute (corresponding to a row) assigned to EmplyeeTable, creates new objects (corresponding to values), and assigns those objects to object attributes (corresponding to columns) and the object attribute corresponding to the row. Assume the columns Name, Phone, SSN, and Salary already exist and are assigned to the object attribute EmployeeTable.

```
Insert_Row_in_EmployeeTable(row, name, phone, ssn, salary)
{   CreateOAinOA(row, EmployeeTable)
    CreateOinOA(name, row)
    Assign(name, Name)
    CreateOinOA(phone, row)
    Assign(phone, Phone)
    CreateOinOA(ssn, row)
    Assign(ssn, SSN)
    CreateOinOA(salary, row)
    Assign(salary, Salary)
}
```

Although the Insert routine applies to the object attributes corresponding to the example schema of figure 3, a similar and corresponding routine could automatically be created for each table of an RDBMS schema, or a generic Insert routine could exist that uses a template specific to each table.

An administrative Delete routine could be used to delete an object attribute, objects and assignments corresponding to a RDMBS row, and column values. Consider, for example the following routine in the context of figure 3(b):

```
Delete_Row_from_EmployeeTable(row)
{   For each object obj in row {
      DeleteO (obj) /*includes deletion of assignments of obj*/
    }
    DeleteOAinOA(row, EmployeeTable) /*includes deletion
          of assignments row to EmployeeTable*/
}
```

Similar to Insert, a Delete routine could automatically be created for each table of an RDBMS schema, or a generic Delete routine could exist that uses a template specific to each table.

Administrative routines not only allow for consistence between RDBMS rows and corresponding NGAC object attributes, objects, and assignments, but also provide a means for testing a user's authority to Insert and Delete RDBMS rows.

For Insert:
The algorithm for translating an Insert statement to an NGAC authorization response assumes the existence of an NGAC administrative Insert routine. The algorithm is as follows:

(1) Invoke the routine corresponding to the table specified in the Insert statement using the identity of the user that issued the Insert statement with the specified row and column values, thereby creating an object attribute that corresponds to the row as well as objects that represent and correspond to column values that are assigned to the row and are appropriately assigned to object attributes that correspond to columns.
(2) If the routine successfully executes, GRANT the SQL Insert statement. Otherwise, DENY access.

For Delete:
The algorithm for translating a Delete statement to an NGAC authorization response assumes the existence of an NGAC administrative Delete routine, particularized for the referenced table. The algorithm is as follows:

(1) Identify the set of rows in the SQL database that meet the criteria included in the Delete SQL statement.
(2) For each row identified in (1), sequentially invoke, using the identity of the user that issued the statement, the Delete routine of the table specified in the Delete statement by caching the parameters of the object attribute corresponding to the identified row and the objects contained in the object attribute.
(3) If any invocation of the routine fails to successfully execute, DENY the SQL Delete statement, and roll back changes due to previous invocations by applying the cache as NGAC administrative Insert routine parameters. Otherwise, GRANT.

6. Related Work

NDAC is not the only system for enforcing fine-grain access control policies over database queries in support of applications. Two others are Oracle's Real Application Security (RAS) [8] and Axiomatics' Data Access Filter (ADAF) [9]. Both are designed to intercept and modify SQL statements for the purpose of applying rule-based controls in database access scenarios.

RAS Allows application developers to define a data security policy, application roles, and application users. At the application layer, security policies are defined in terms of Access Control Lists on dynamically created (using a "where" clause) Data Realms (set of rows) and static "Views" on columns using the RAS API. In effect, control is provided down to the record/field level.

ADAF includes a proxy that intercepts SQL statements, which in turn are sent to an ADAF engine. The engine employs two policy enforcing capabilities. First, a "where" clause is computed and added to the SQL statement, thereby filtering out rows for which the user is not authorized. This filtering operates on XACML 3.0 [10] policies in terms of object attributes created to correspond to the tables and columns of the database schema. Second, ADAF uses Masking to further redact individual cells of the filtered rows, thereby providing filtering down to the record/field level.

NDAC has a number of similarities and differences with RAS and ADAF. The RAS protection scheme is application centric and DBMS specific, while ADAF and NDAC allow the same policies to protect multiple databases from queries sent from multiple applications. ADAF and RAS policies for controlling access to rows are fully dependent on the database schema definition, while NDAC is not. NDAC can define object attributes that contain schema related object-attributes (e.g., Public and Sensitive of figure

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3(b)) and express policies in terms of those object attributes. This is an important distinction because enterprise policies are fluid and change over time while schemas are ridged and typically remain fixed.

In contrast to ADAF and RAS, NDAC does not need to maintain and coordinate policies of two access control schemes to achieve fine-grain access control. Although NDAC is shown external to the DBMS, its policies are expressed in terms of relations like RAS, allowing NDAC to be implemented as a database-kernel loadable module.

Since ADAF is based on XACML, it is not amenable to policy review, while RAS and NDAC can query the rule configuration (relations) to determine the tables, rows, and columns accessible to a given user in advance (without computing a decision). Moreover, NDAC can graphically visualize the overall set of rules. See [7] for NGAC algorithms and techniques for efficient policy review and visualization.

7. CONCLUSION

This paper describes a system that leverages a ANSI/INCITS Next Generation Access Control (NGAC) standard called Next-generation Database Access Control (NDAC) for accessing data in tables, rows, and columns in existing RDBMS products. NDAC imposes access control at the data level and eliminates the need for implementing and managing access control in applications, and/or through the use of proprietary RDBMS mechanisms. As a consequence, the same policies can protect multiple databases from queries sent from multiple applications. Furthermore, NDAC does not only provide control down to the field level, but to the level of varying fields of select rows. Although other technologies achieve a similar granularity of protection through the combined use of multiple protection mechanisms, NDAC is unique in its use of just one policy store. Operationally, users issue wide sweeping queries and NDAC allows access the optimal amount of data permissible for the user.

The NDAC process for expressing access control policies begins with an existing RDBMS schema that includes columns and tables that are automatically converted into NGAC corresponding object attributes and assignments. Since rows are also object containers, existing rows can automatically be converted as well. NGAC data elements and relations also include User Attributes, a generic set of operations, and three types of relations for specifying an access policy. Once the RDBMS schema has been converted, NGAC relations are configured in formulating policy in terms of the created object attributes and assignments using NGAC's administrative API. The resulting data elements and relations are stored as NGAC Access Control Data. In addition to the conversion and the additional data elements and relations, NDAC includes an Access Manager for trapping SQL queries from applications and a Translator for converting SQL queries along with a user identity to NGAC inputs and NGAC authorization responses to those inputs to either an access Deny or permitted SQL queries that are sent to the RDBMS for policy preserving access.

The U.S. Government has filed a patent application of certain aspects of the subject matter disclosed in this paper.

Disclaimer: *Products may be identified in this document, but identification does not imply recommendation or endorsement by NIST, nor that the products identified are necessarily the best available for the purpose.*

8. REFERENCES

[1] Information technology - Next Generation Access Control - Functional Architecture (NGAC-FA), INCITS 499-2013, American National Standard for Information Technology, American National Standards Institute, March 2013.

[2] American National Standards Institute, Information technology – Next Generation Access Control – Generic Operations and Data Structures (GOADS), INCITS 526-2016, American National Standard for Information Technology, January 2016.

[3] NIST Policy Machine Versions 1.5 and 1.6 - Harmonia [Website], https://github.com/PM-Master [accessed 9/26/16].

[4] D. Ferraiolo, S. Gavrila, and W. Jansen, "Policy Machine: features, architecture, and specification," National Institute of Standards and Technology, Internal Report 7987 Rev. 1, 2015.

[5] D. Ferraiolo, S. Gavrila, and W. Jansen, "On the Unification of Access Control and Data Services," in *Proceedings of the 2014 IEEE 15th International Conference of Information Reuse and Integration*, IEEE, 2014, pp. 450 – 457. http://dx.doi.org/10.1109/IRI.2014.7051924

[6] D.F. Ferraiolo, V. Atluria, and S.I. Gavrila, "The Policy Machine: A Novel Architecture and Framework for Access Control Policy Specification and Enforcement," *Journal of Systems Architecture*, vol. 57, no. 4, pp. 412-424, April 2011. http://dx.doi.org/10.1016/j.sysarc.2010.04.005

[7] P. Mell, J. Shook, S. Gavrila, Restricting Insider Access through Efficient Implementation of Multi-Policy Access Control Systems. In *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*. Vienna, Austria, October 24-26, 2016.

[8] http:/www.oracle.com/technetwork/database/security/real-application-security/real-application-security-1964775.html.

[9] https:/www.axiomatics.com/resources/102-data-sheets/431-axiomatics-data-access-md-filter-data-sheet.html.

[10] The Xtensible Access Control Markup Language (XACML), Version 3.0, OASIS Standard, January 22, 2013. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

HMFEv - An Efficient Multivariate Signature Scheme

Albrecht Petzoldt[1], Ming-Shing Chen[2], Jintai Ding[3], Bo-Yin Yang[2]

[1] National Institute for Standards and Technology, Gaithersburg, Maryland, USA
[2] Academia Sinica, Taipei, Taiwan
[3] University of Cincinnati, Ohio, USA
albrecht.petzoldt@nist.gov,jintai.ding@gmail.com,{mschen,byyang}@crypto.tw

Abstract. Multivariate Cryptography, as one of the main candidates for establishing post-quantum cryptosystems, provides strong, efficient and well-understood digital signature schemes such as UOV, Rainbow, and Gui. While Gui provides very short signatures, it is, for efficiency reasons, restricted to very small finite fields, which makes it hard to scale it to higher levels of security and leads to large key sizes.
In this paper we propose a signature scheme called HMFEv ("Hidden Medium Field Equations"), which can be seen as a multivariate version of HFEv. We obtain our scheme by applying the Vinegar Variation to the MultiHFE encryption scheme of Chen et al.. We show both theoretically and by experiments that our new scheme is secure against direct and Rank attacks. In contrast to other schemes of the HFE family such as Gui, HMFEv can be defined over arbitrary base fields and therefore can be much more efficient in terms of both performance and memory requirements. Our scheme is therefore a good candidate for the upcoming standardization of post-quantum signature schemes.

Keywords: Post-Quantum Cryptography, Multivariate Cryptography, Signature Schemes, NIST Call for Proposals

1 Introduction

Multivariate Public Key Cryptosystems (MPKCs) are one of the main candidates for guaranteeing the security of communication in a quantum world [1]. Multivariate schemes are in general very fast and require only modest computational resources, which makes them attractive for the use on low cost devices like smart cards and RFID chips [4,6]. Additionally, at least in the area of digital signatures, there exists a large number of practical multivariate schemes.
The existing multivariate signature schemes can be divided into two main groups. The first are the SingleField schemes UOV and Rainbow, which follow the same type of design strategy using Oil-Vinegar polynomials. We believe that these two schemes are more or less the best which can be achieved from this fundamental design.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

On the other hand, we have the BigField schemes HFEv- and Gui, which combine the HFE design with the Minus and Vinegar modifiers. These schemes make use of an HFE polynomial, whose degree D is very much affected by the size of the underlying field. We believe that, for security reasons, this degree should be chosen at least q^2+1 , where q is the cardinality of the underlying field. However, during the signature generation process, we have to invert this univariate HFE polynomial and the complexity of this step can be estimated by $\mathcal{O}(D^3)$. To solve this conflict between security and efficiency, we have to build the scheme over very small finite fields such as GF(2) and GF(4). However, in this case, we have to choose the number of variables to be large, which leads to large key sizes and makes the scheme less efficient. Therefore it is a natural question, if it is possible to use large base fields such as GF(31) or GF(256) for the design of multivariate signature schemes of the HFEv- type.

In 2008, Chen et al. proposed a multivariate encryption scheme called MultiHFE [7], which can be seen as a multivariate version of HFE. While the scheme is very efficient, its security appeared to be weak and it was broken by Bettale et al. [3] by a generalization of the Kipnis-Shamir attack against HFE using the MinRank property of the system.

In this paper, we propose a signature scheme called HMFEv ("Hidden Medium Field Equations"), which we obtain by applying the Vinegar modification to MultiHFE. We show both theoretically and by experiments that our scheme is secure against direct and Rank attacks of the Kipnis-Shamir / Bettale type and analyze the security of our scheme against other known attacks against multivariate schemes, including differential attacks and Hashimotos attack against the MultiHFE encryption scheme. Our scheme can be seen as an extension of the Gui and QUARTZ signature schemes. However, by enabling a flexible choice of the base field, our new scheme overcomes a fundamental practical problem in the HFEV- design. While Gui and QUARTZ are, for efficiency reasons, mainly restricted to the field GF(2), our scheme allows the choice of an arbitrary base field. This allows us to reduce the number of equations and variables in the public system significantly, which leads to smaller key sizes and more efficient signature generation and verification processes. Furthermore, this enables an easy scalability of our scheme to higher levels of security. Our scheme is therefore a very strong candidate for the upcoming standardization of post-quantum signature schemes.

The rest of this paper is organized as follows. Section 2 gives an overview of the basic concepts of multivariate cryptography. In Section 3 we describe the MultiHFE encryption scheme which is the basis of our construction and analyze its security and efficiency. Section 4 describes our new HMFEv signature scheme in detail. In Section 5 we analyze the security of our scheme, in particular its behavior against direct and rank attacks. Section 6 proposes concrete parameter sets for our scheme for different levels of security. Section 7 compares our HMFEv scheme with other multivariate signature schemes of the HFEv- type, in particular Gui, and Section 8 concludes the paper.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2 Multivariate Cryptography

The public key of a multivariate public key cryptosystem (MPKC) is a set of multivariate quadratic polynomials. The security of these schemes is based on the MQ Problem of solving such a system. The MQ problem (for $m \approx n$) is proven to be NP-hard even for quadratic polynomials over the field GF(2) [14] and believed to be hard on average (both for classical and quantum computers). To build a public key cryptosystem based on the MQ problem, one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ (central map). To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine (or linear) maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. The *public key* of the scheme is therefore given by $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$. The *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert the public key.
In this paper we concentrate on multivariate schemes of the MediumField familiy. For this type of schemes, one chooses two integers k and ℓ and sets $n = k \cdot \ell$. The central map $\mathcal{F}$ of the scheme is a specially chosen easily invertible polynomial map over the vector space $\mathbb{E}^k$, where $\mathbb{E}$ is a degree ℓ extension field of $\mathbb{F}$. Using an isomorphism $\phi : \mathbb{F}^\ell \to \mathbb{E}$ we can transform $\mathcal{F}$ into a map

$$\bar{\mathcal{F}} = \underbrace{(\phi^{-1} \times \cdots \times \phi^{-1})}_{k-\text{times}} \circ \mathcal{F} \circ \underbrace{(\phi \times \cdots \times \phi)}_{k-\text{times}} : \mathbb{F}^n \to \mathbb{F}^n. \tag{1}$$

from $\mathbb{F}^n$ to itself. The map $\mathcal{F}$ is chosen in such a way that the map $\bar{\mathcal{F}}$ consists of multivariate quadratic polynomials. The *public key* has the form $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$ with two invertible affine maps $\mathcal{S}, \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$, the *private key* consists of $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$.

3 The MultiHFE scheme

An important example for a multivariate scheme from the MediumField family is the MultiHFE scheme of Chen et al. [7]. In its basic version, the scheme can be used both as an encryption and signature scheme.

The k components $\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(k)}$ of the central map $\mathcal{F}$ are of the form

$$\mathcal{F}^{(i)} = \sum_{1 \le r \le s \le k} \alpha_{rs}^{(i)} \cdot X_r X_s + \sum_{1 \le r \le s} \beta_r^{(i)} \cdot X_r + \gamma^{(i)} \quad (i = 1, \ldots, k) \tag{2}$$

with coefficients $\alpha_{rs}^{(i)}$, $\beta_r^{(i)}$ and $\gamma^{(i)} \in \mathbb{E}$. Note that the polynomials $\mathcal{F}^{(i)}$ $(i = 1, \ldots, k)$ are multivariate polynomials of the HFE type with $D = 2$. The map $\bar{\mathcal{F}}$ of the MultiHFE signature scheme is defined as shown in equation (1) and is, due to the Frobenius isomorphism, a multivariate quadratic map over the vector space $\mathbb{F}^n$. To hide the structure of $\bar{\mathcal{F}}$ in the public key, one composes it with two invertible affine maps $\mathcal{S}$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. Therefore, the *public key* of the scheme is given by $\mathcal{P} = \mathcal{S} \circ \bar{\mathcal{F}} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$, the *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Signature Generation: In order to generate a signature for a message d one uses a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^n$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^n$ and performs the following three steps.

1. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^n$ and lift the result to the vector space $\mathbb{E}^k$. Denote the result by $\mathbf{X}$.
2. Invert the central map $\mathcal{F}$ to obtain $\mathbf{Y} = \mathcal{F}^{-1}(\mathbf{X}) \in \mathbb{E}^k$ and compute $\mathbf{y} = (\phi^{-1} \times \cdots \times \phi^{-1})(\mathbf{Y}) \in \mathbb{F}^n$. Since $\mathcal{F}$ is a system of k randomly chosen quadratic polynomials in k variables, we need for this step a system solver like XL [24] or a Gröbner Basis algorithm such as F_4 [13] or F_5.
3. Compute the signature $\mathbf{z} \in \mathbb{F}^n$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Verification: To check, if $\mathbf{z} \in \mathbb{F}^n$ is indeed a valid signature for a message d, one computes the hash value $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

3.1 Efficiency

The most complex step during the decryption process of MultiHFE is the solution of the multivariate quadratic system $\mathcal{F}(Y_1, \ldots, Y_k) = (X_1, \ldots, X_k)$ (k equations in k variables) over the extension field $\mathbb{E}$. Since the coefficients of the system $\mathcal{F}$ are chosen randomly, this step has to be performed by a system solver like XL [24] or a Gröbner Bases algorithm such as F_4 [13]. If the number k of equations and variables in this system is small, these algorithms can invert $\mathcal{F}$ very efficiently. However, when the parameter k gets larger, the decryption process of MultiHFE becomes very costly and the scheme therefore gets inefficient.

3.2 The Rank Attack against HFE and MultiHFE

In [17], Kipnis and Shamir proposed a rank based attack against the univariate HFE scheme. The key idea of this attack is to lift all the maps $\mathcal{S}$, $\mathcal{P}$ and $\mathcal{T}$ to univariate maps $\mathcal{S}^\star$, $\mathcal{P}^\star$ and $\mathcal{T}^\star$ over the extension field $\mathbb{E}$. Since the rank of the central map $\mathcal{F}$ is bounded from above by $r = \lfloor \log_q(D-1) \rfloor + 1$, this enabled them to recover the private key by solving an instance of a MinRank problem. However, since computing the map $\mathcal{P}^\star$ appeared to be very costly, the attack of Kipnis and Shamir is not very efficient.
Later, Bettale et al. [3] found a way to perform the attack of Kipnis and Shamir without the need of recovering the map $\mathcal{P}^\star$. Besides improving the efficiency of the Kipnis-Shamir attack, this makes it much easier to extend the attack to MultiHFE. Due to lack of space we cannot present all the details of the attacks of Kipnis-Shamir and Bettale here and refer to the papers [17], [3] and the extended version of this paper for a detailed analysis of the attacks. Here, we just present the main results of [3].

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Theorem 1. *For MultiHFE, recovering the affine transformation $\mathcal{T}$ reduces to simultaneously solving k MinRank problems over the base field.*

With this, Bettale et al. could further prove

Theorem 2. *The complexity of solving the MultiHFE MinRank problem is $\mathcal{O}(\ell^{(k+1)\omega})$ with $2 < \omega \leq 3$ being the linear algebra constant and ℓ being the degree of the field extension $\mathbb{E}|\mathbb{F}$.*

We therefore face the following problem: If the parameter k in MultiHFE is small, the scheme can be easily broken by the MinRank attack. On the other hand, if we choose k larger, the efficiency of the scheme becomes quite bad.
In the following we show how to solve this dilemma by modifying the MultiHFE scheme.

4 The new Signature Scheme HMFEv

Let $\mathbb{F}$ be a finite field and k, ℓ and v be integers. We set $n = k \cdot \ell$. Furthermore, let $g(X) \in \mathbb{F}[X]$ be an irreducible polynomial of degree ℓ and $\mathbb{E} = \mathbb{F}[X]/g(X)$ the corresponding extension field. We define an isomorphism $\phi : \mathbb{F}^\ell \to \mathbb{E}$ by

$$\phi(x_1, \ldots, x_\ell) = \sum_{i=1}^{\ell} x_i \cdot X^{i-1}.$$

The *central map* $\mathcal{F} : \mathbb{E}^k \times \mathbb{F}^v \to \mathbb{E}^k$ of the scheme consists of k components $\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(k)}$ of the form

$$\mathcal{F}^{(i)} = \sum_{r,s=1}^{k} \alpha_{rs}^{(i)} \cdot X_r X_s + \sum_{r=1}^{k} \beta_r^{(i)}(v_1, \ldots, v_v) \cdot X_r + \gamma^{(i)}(v_1, \ldots, v_v)$$

with coefficients $\alpha_{rs}^{(i)} \in \mathbb{E}$, linear functions $\beta_r^{(i)} : \mathbb{F}^v \to \mathbb{E}$ and quadratic maps $\gamma^{(i)} : \mathbb{F}^v \to \mathbb{E}$ ($i \in \{1, \ldots, k\}$).
Due to the special form of $\mathcal{F}$, the map

$$\bar{\mathcal{F}} = \underbrace{(\phi^{-1} \times \cdots \times \phi^{-1})}_{k-\text{times}} \circ \mathcal{F} \circ (\underbrace{\phi \times \cdots \times \phi}_{k-\text{times}} \times \mathrm{id}_v)$$

is a multivariate quadratic map from $\mathbb{F}^{n+v}$ to $\mathbb{F}^n$. Here, id_v is the identity map over the vector space $\mathbb{F}^v$.
To hide the structure of $\bar{\mathcal{F}}$ in the public key, we combine it with two randomly chosen invertible affine maps $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^n$ and $\mathcal{T} : \mathbb{F}^{n+v} \to \mathbb{F}^{n+v}$.
The *public key* of the scheme is given by

$$\mathcal{P} = \mathcal{S} \circ \bar{\mathcal{F}} \circ \mathcal{T} : \mathbb{F}^{n+v} \to \mathbb{F}^n,$$

the *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$.

Signature Generation: To generate a signature for a document d, we use a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^n$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^n$. After that, we perform the following six steps

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

1. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w})$.
2. Lift the result to the extension field $\mathbb{E}$ by computing $X_i = \phi(x_{(i-1)\cdot\ell+1}, \ldots, x_{i\cdot\ell})$ $(i = 1, \ldots, k)$.
3. Choose random values for the Vinegar variables $v_1, \ldots, v_v \in \mathbb{F}$ and substitute them into the central map components to obtain the parametrized maps $\mathcal{F}_V^{(1)}, \ldots, \mathcal{F}_V^{(k)}$.
4. Use the XL-Algorithm or a Gröbner basis method to compute $Y_1, \ldots, Y_k$ such that $\mathcal{F}_V^{(i)}(Y_1, \ldots, Y_k) = X_i$ $(i = 1, \ldots, k)$.
5. Move the result down to the vector space by computing $\mathbf{y} = (\phi^{-1}(Y_1), \ldots, \phi^{-1}(Y_k), v_1, \ldots, v_v) \in \mathbb{F}^{n+v}$.
6. Compute the signature $\mathbf{z} \in \mathbb{F}^{n+v}$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Signature Verification: In order to check, if $\mathbf{z} \in \mathbb{F}^{n+v}$ is indeed a valid signature for the document d, one computes $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

5 Security

In this Section we analyze the security of our scheme. In particular we study both theoretically and using computer experiments the behavior of our scheme against direct and rank attacks. .

5.1 Direct and Rank attacks

The complexity of a direct attack is closely related to the degree of regularity of the system. Therefore the key task is to study the degree of regularity of a direct attack against our scheme.
From the work of Ding and Hodges in Crypto 2011 [11] we know that the degree of regularity of a direct attack against an HFE scheme can be estimated by looking at a single polynomial in the extension field $\mathbb{E}$, and the rank of the associated quadratic form.
In the case of HMFEv, the situation is slightly different, but still very similar. For HMFEv, the components of the public key come from several polynomials over the medium field, which are given as

$$\mathcal{F}^{(i)} = \sum_{r,s=1}^{k} \alpha_{rs}^{(i)} \cdot X_r X_s + \sum_{r=1}^{k} \beta_r^{(i)}(v_1, \ldots, v_v) \cdot X_r + \gamma^{(i)}(v_1, \ldots, v_v) \quad (1 \leq i \leq k).$$

Using the same argument as in the work of Ding and Yang in [10] we can, under the assumption of $v \leq \ell$, lift each map $\mathcal{F}^{(i)}$ $(1 \leq i \leq k)$, which is a map from $\mathbb{E}^k \times \mathbb{F}^v$ to $\mathbb{E}$, to a map $\mathcal{F}'^{(i)}$ from $\mathbb{E}^{k+1}$ to $\mathbb{E}$. Here, the additional component in the domain comes from the use of the vinegar variables. Then we can look at the rank of the quadratic form associated to the polynomial $\mathcal{F}'^{(i)}$ as in the case of the original Kipnis-Shamir attack.
Using the same method as in [10] we can prove

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Theorem 3. *If $v \leq \ell$ holds, the rank of the quadratic form associated to $\mathcal{F}'^{(i)}$ is greater or equal to $k + v$.*

The proof follows directly from that in [10].
This theorem directly gives us a lower bound for the complexity of the MinRank attack (see Theorem 2) by

$$\text{Complexity}_{\text{MinRank}} \geq \ell^{(k+v+1)\cdot\omega}. \tag{3}$$

Theorem 3 allows us to use the method of [11] to derive directly

Theorem 4. *The degree of regularity of a direct attack against an HMFEv system is, under the assumption of $v \leq \ell$, upper bounded by*

$$d_{\text{reg}} \leq \begin{cases} \frac{(q-1)(k+v-1)}{2} + 2 & \text{if } q \text{ even and } (k+v) \text{ odd} \\ \frac{(q-1)\cdot(k+v)}{2} + 2 & \text{otherwise} \end{cases}. \tag{4}$$

Equation (4) gives an upper bound for the degree of regularity of a direct attack against our scheme. However, in order to estimate the security of the HMFEv-scheme in practice, we need to analyze if the bound given by (4) is reasonably tight. Furthermore we want to study, if, as equation (4) indicates, only the sum and not the concrete choice of k and v determines the degree of regularity of a direct attack against an HMFEv system. To answer these two questions, we performed a large number of experiments.
Our experiments (see in Section A of the appendix of this paper) show that the upper bound on the degree of regularity given by equation (4) is relatively tight. We could find several MHFEv instances which actually meet the upper bound and found that in most other cases the upper bound is missed only by one. Regarding the second question, we found that the concrete choice of k and v has no influence on the behavior of the scheme against direct attacks as long as v is not too small.
The experiments in the appendix deal with HMFEv schemes over very small fields such as GF(2) and GF(3). However, one major benefit of the HMFEv scheme is that, in contrast to HFEv-, it can be efficiently used over fields of arbitrary size. As our experiments (see Section 6) show, these systems behave much more like random systems and we can reach high degrees of regularity, by which we can show the security of our scheme against direct attacks.

5.2 Quantum Attacks

In [22] Schwabe and Westerbaan showed that a binary system of n multivariate quadratic equations can be solved by a quantum computer in time $2^{n/2} \cdot 2 \cdot n^2$. Since our systems over GF(256) can easily be translated into systems over GF(2), this attack affects also our scheme (at least in theory). However, since this transition increases the number of variables in the system by a factor of 8, we do not have to consider this type of attack here.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

5.3 Other Attacks and A Remark on the Minus Method

Additional to direct, quantum and rank attacks, we analyzed the security of our scheme against other known attacks against multivariate schemes, including differential attacks and Hashimotos attack against the original MultiHFE encryption scheme [15] and found that these attacks do not apply against our scheme. However, due to lack of space, we can not present the details of our analysis here and refer to the extended version of this paper.

Remark. A natural question here is, why we do not use the Minus method as in the case of HFEv-. There are two main reasons.
1. In opposite to the Vinegar variation, the Minus modification does not help to defend our scheme against Hashimotos attack against the original MultiHFE encryption scheme [15].
2. If we use the above method to analyze the MinRank attack, we can prove that the MinRank should be $k + v + ak$, where a is the number of Minus equations. But there appears a new and very interesting problem regarding the degree of regularity. If we follow our usual method, we derive

$$d_{\text{reg}} \leq \begin{array}{ll} \frac{(q-1)(k+v+ak-1)}{2} + 2 & \text{if } q \text{ even and } (k+v+ak) \text{ odd} \\ \frac{(q-1)\cdot(k+v+ak)}{2} + 2 & \text{otherwise} \end{array} . \qquad (5)$$

However our experiments show that this bound is not tight. This can be explained as follows. In the case of HFEv-, the estimate comes from using a single polynomial on a large field, and a single polynomial already determines the whole system; in the case of MHFEv-, the system is determined by k polynomials, not by one; since our analysis considers only one of these polynomials, it does not use all the information available and therefore overestimates the degree of regularity. This means we have a gap in the knowledge on estimating the degree of regularity in MHFEv-, which is the reason we propose the MHFEv system (i.e. only with Vinegar). This problem is very interesting and important, and we are going to deal with it in a subsequent paper.

6 Parameter Choice

In this section we consider the question how to find good parameter sets for our scheme. In particular, we aim at finding parameters for HMFEv over the fields GF(31) and GF(256).[4]

[4] The reason why we do not propose parameters for our scheme over GF(16) is the following: To defend the scheme against the quantum attack (see Section 5.2), we need a large number of equations over GF(16). This actually makes the schemes less efficient than HMFEv over GF(31) or GF(256).

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

6.1 How to choose the parameter k?

The first question we have to answer in order to find suitable parameters for our scheme is how to choose the parameter k and therefore the number of components of the central map. Reducing the value of k will speed up the signature generation process of our scheme since it decreases the size of the multivariate quadratic system we have to solve. However, if k is too small, this might bring the security of our scheme into jeopardy.
For fields of odd characteristic (e.g. $\mathbb{F}$=GF(31)) we choose the parameter k to be 2. However, in order to increase the security of our scheme against Rank attacks, we choose in this case the components of the central map $\mathcal{F}$ in a special way. Let F_1 and F_2 be the 2×2 matrices representing the homogeneous quadratic parts of the maps $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$. A linear combination of F_1 and F_2 of rank 1 exists if and only if the quadratic polynomial $p(X) = \det(F_1 + X \cdot F_2) \in \mathbb{E}[X]$ has a solution. We therefore choose the coefficients of $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ in such a way that the polynomial $p(X)$ is irreducible.
For fields of even characteristic, the symmetric matrices representing the quadratic maps $\mathcal{F}^{(i)}$ contain zero elements on the main diagonal. Therefore, for $k = 2$, the rank of these matrices would be 1 and the upper linear combination of the maps $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ would actually lead to a matrix of rank 0 (i.e. no quadratic terms at all.) To prevent this, we choose for fields of even characteristic the parameter k to be 3.

6.2 Experiments with direct attacks against HMFEv schemes over GF(31) and GF(256)

In Section 5.1 we already presented some results of experiments with the direct attack against HMFEv instances. However, in Section 5.1, we looked at HMFEv schemes over very small fields, for which the bound given by equation (4) is more or less tight. In this section we consider the question if concrete instances of HMFEv over the larger fields GF(31) and GF(256) are hard to solve.
To do this, we created for different parameter sets HMFEv systems over GF(31) and GF(256) and solved these systems, after fixing v variables to obtain a determined system, with the F_4 algorithm integrated in MAGMA. The experiments were performed on a single core of a server with 16 AMD Opteron processors (2.4 GHz) and 128 GB of RAM. For each parameter set we performed 10 experiments. Table 1 shows the results.

As the table shows, we can, for HMFEv instances over both GF(31) and GF(256), reach high degrees of regularity. In particular we see that, for the parameter sets proposed in the next section, the degree of regularity of a direct attack is at least 17. By substituting this value into the formula

$$\text{Complexity}_{\text{direct attack}} \approx 3 \cdot \binom{n + d_{\text{reg}}}{d_{\text{reg}}}^2 \cdot \binom{n}{2} \tag{6}$$

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

GF(31)	parameters (k, ℓ, v)	(2,6,4)	(2,7,4)	(2,8,4)	random
	m,n	12,12	14,14	16,16	16,16
	d_{reg}	14	16	18	18
	time	1,906	164,110	-	-
	memory (MB)	949	17,165	ooM	ooM
GF(256)	parameters (k, ℓ, v)	(3,3,6)	(3,4,6)	(3,5,6)	random
	m,n	9,9	12,12	15,15	15,15
	d_{reg}	11	14	17	17
	time	4.0	1,875	-	-
	memory (MB)	24.5	949	ooM	ooM

Table 1. Experiments with the direct attack against HMFEv schemes over GF(31) and GF(256)

we find that the complexity of a direct attack against the HMFEv instances shown in Table 2 is beyond the claimed levels of security.
Also note that, for the underlying fields of GF(31) and GF(256), the public systems of HMFEv behave very similar to random systems. This also holds when guessing some variables before applying the F_4 algorithm (hybrid approach).

6.3 Parameters

Table 2 shows, for different levels of security (128, 192, and 256 bit) our parameter recommendations for the HMFEv signature scheme over GF(31) and GF(256). In the case of GF(31), we store one element of GF(31) in 5 bits, while 24 bits can be efficiently stored in 5 GF(31) elements.

security level (bit)	parameters $(\mathbb{F}, k, \ell, v)$	public key size (kB)	private key size (kB)	hash size (bit)	signature size (bit)
128	(GF(31),2,28,12)	81.8	8.9	277	337
	(GF(256),3,15,16)	85.8	15.2	360	488
192	(GF(31),2,40,17)	234.7	20.0	396	481
	(GF(256),3,23,21)	282.1	35.0	552	720
256	(GF(31),2,55,21)	583.9	38.0	544	649
	(GF(256),3,31,26)	659.4	65.3	744	952

Table 2. Parameter Recommendations for the HMFEv Signature Scheme

The parameter sets given in Table 2 are chosen in such a way that the complexities of direct attacks (including hybrid approach; see Section 6.2) and Rank attacks (see equation (3)) against the given HMFEv instances are beyond the claimed levels of security. To be on the conservative side we chose, in the formula

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

(3), the linear algebra constant ω to be 2. Furthermore, in the case of MHFEv over GF(31), we had to take care of the fact that the public systems contain enough equations to prevent collision attacks against the hash function.

7 Comparison

We briefly describe our implementation in the Appendices B and C of this paper.

The basic idea of the HMFEv signature scheme is very similar to that of Gui [20]: by applying the Vinegar modification it is possible to increase both the security and the efficiency of the scheme significantly. However, there are at least three major advantages of our scheme compared to Gui.

First, for efficiency reasons, the Gui signature scheme is restricted to the field GF(2). This leads to a large number of variables in the scheme and therefore to large key sizes. . On the other hand, the HMFEv signature scheme can be defined over any field. This enables us to decrease the number of variables in the system and therefore reduces the public key size of the scheme significantly (see Table 3).

Secondly, for the parameter sets recommended in [20], the output size of the HFEv- public key is only 90 bit. Therefore, in order to defend the HFEv- signature scheme against collision attacks, the authors of Gui had to create a specially designed signature generation process for their scheme which inverts the HFEv- core map several times. Since the design of Gui requires the single HFEv- systems to have exactly one solution, generating one single Gui signature implies about 11 inversions of the HFEv- map, which leads to a relatively low performance of Gui. In the case of the HMFEv scheme, we do not need this multiple inversion of the core map, which makes the signature generation process of our scheme much faster and easier to implement. Furthermore, since the number of variables in the public systems of Gui is much larger than for our scheme, the evaluation of the HMFEv public systems and therefore the verification process of our scheme is much cheaper.Table 3 compares, for a security level of 80 bit, the HMFEv and Gui signature schemes with respect to key and signature sizes as well as the running time of the signature generation and verification process. Note that, for higher levels of security, the benefits of our scheme would be even more significant. The schemes listed in the table run on an Intel Xeon E3-1245 processor with 3.4 GHz. The parameters and running times in the first three rows of the table are taken from the paper [20]. The third major advantage of the HMFEv scheme is that, in contrast to other HFEv- based schemes like Gui, the scheme can be scaled much easier to higher levels of security. For example, in order to obtain a quantum security level of 256 bit, we need an internal state of at least 480 bit (c.f. Section 5.2), which means that we need at least 480 variables over GF(2). This would lead to key sizes which are completely impractical. In the case of HMFEv-, we can increase the size of the internal state simply by choosing a larger base field, which has far less influence on key sizes.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

	public key size (kB)	private key size (kB)	signature size (bit)	sign. gen. time (ms)	verification time (ms)
Gui (GF(2),96,5,6,6)	61.6	3.1	126	0.07	0.02
Gui(GF(2),95,9,5,5)	59.2	3.0	120	0.18	0.02
Gui(GF(2),94,17,4,4)	56.8	2.9	124	0.73	0.02
HMFEv (GF(31),2,18,8)	22.5	3.5	218	0.20	0.012
HMFEv (GF(256),3,9,12)	21.6	6.0	312	0.24	0.02
HMFEv (GF(31),2,28,12)	81.8	8.9	337	0.40	0.04
HMFEv (GF(256),3,15,16)	85.8	15.2	488	0.36	0.05

Table 3. Comparison of HMFEv and Gui (80 bit security)

8 Conclusion

In this paper we proposed a new multivariate signature scheme called HMFEv which is obtained by applying the Vinegar modification to the MultiHFE scheme of Chen et al. [7]. By using this variation, we are able to reduce the number of components in the central map of the scheme and therefore to increase the efficiency significantly. We studied the security of our scheme against direct and rank attacks both theoretically and experimentally and showed that our scheme can not be attacked using differential methods or Hashimotos attack against the original MultiHFE scheme. We showed that our scheme is much more efficient than the Gui signature scheme with regard to key sizes, performance and scalability. Future work includes in particular further optimization of the implementation to enable a better comparison of our results with those from [20] as well as a careful study on the effects of applying the Minus modification on HMFEv.

Disclaimer

Certain algorithms and commercial products are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the algorithms or products identified are necessarily the best available for the purpose.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

References

1. D.J. Bernstein, J. Buchmann, E. Dahmen (eds.): Post Quantum Cryptography. Springer, 2009.
2. L. Bettale, L.C Faugère, L. Perret: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology 3, pp. 177-197 (2009).
3. L. Bettale, J.C. Faugère, L. Perret: Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. Des. Codes Cryptography 69 (1), pp. 1–52 (2013).
4. A. Bogdanov, T. Eisenbarth, A. Rupp, C. Wolf: Time-area optimized public-key engines: MQ-cryptosystems as replacement for elliptic curves? CHES 2008, LNCS vol. 5154, pp. 45-61. Springer, 2008.
5. R. Cartor, R. Gipson, D. Smith-Tone, J. Vates: On the Differential Security of the HFEv- Signature Primitive. PQCrypto 2016, LNCS vol. 9606, pp. 162 - 181. Springer, 2016.
6. A.I.T. Chen, M.-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E. L.-H. Kuo, F. Y.-S. Lee, B.-Y. Yang: SSE implementation of multivariate PKCs on modern x86 cpus. CHES 2009, LNCS vol. 5747, pp. 33 - 48. Springer, 2009.
7. C.H.O. Chen, M.S. Chen, J. Ding, F. Werner, B.Y. Yang: Odd-char multivariate Hidden Field Equations. IACR eprint, http://eprint.iacr.org/2008/543 (2008).
8. T. Daniels, D, Smith-Tone: Differential Properties of the HFE Cryptosystem. PQCrypto 2016, LNCS vol. 8772, pp. 59 - 75. Springer, 2014.
9. J. Ding, J. E. Gower, D. S. Schmidt: Multivariate Public Key Cryptosystems. Springer, 2006.
10. J. Ding, B.Y. Yang: Degree of regularity for HFEv and HFEv-. PQCrypto 2013, LNCS vol. 7932, pp. 52 - 66. Springer, 2013.
11. J. Ding, T. Hodges: Inverting HFE is quasipolynomial for all fields. CRYPTO 2011, LNCS vol. 6841, pp. 724 - 742. Springer, 2011.
12. J. Ding, D. S. Schmidt: Rainbow, a new multivariate polynomial signature scheme. ACNS 2005, LNCS vol. 3531, pp. 164-175. Springer, 2005.
13. J.C. Faugère: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra 139, pp. 61-88 (1999).
14. M. R. Garey and D. S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company 1979.
15. Y. Hashimoto: Cryptanalysis of Multi HFE. IACR eprint, http://eprint.iacr.org/2015/1160.pdf (2015).
16. A. Kipnis, L. Patarin, L. Goubin: Unbalanced Oil and Vinegar Schemes. EUROCRYPT 1999, LNCS vol. 1592, pp. 206–222. Springer, 1999.
17. A. Kipnis, A. Shamir: Cryptanalysis of the HFE Public Key Cryptosystem. CRYPTO 99, LNCS vol. 1666, pp. 19 - 30. Springer 1999.
18. J. Patarin, N. Courtois, L. Goubin: QUARTZ, 128-Bit Long Digital Signatures. CTRSA 2001, LNCS vol. 2020, pp. 282-297. Springer, 2001.
19. J. Patarin: Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms. EUROCRYPT 1996, LNCS vol. 1070, pp. 33 - 48. Springer 1996.
20. A. Petzoldt, M.S. Chen, B.Y. Yang, C. Tao, J. Ding: Design Principles for HFEv-based Signature Schemes. ASIACRYPT 2015 - Part 1, LNCS vol. 9452, pp. 311-334. Springer, 2015.
21. R. L. Rivest, A. Shamir, L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 21 (2), pp. 120-126 (1978).

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

22. P. Schwabe, B. Westerbaan: Solving binary MQ with Grovers algorithm. Available at https://cryptojedi.org/papers/mqgrover-20160901.pdf.
23. P. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. Comput. 26 (5), pp. 1484 - 1509 (1997).
24. B.-Y. Yang, J.-M. Chen: Theoretical Analysis of XL over Small Fields. ACISP 2004, LNCS vol. 3108, pp.277-288. Springer 2004.

A Results of our Computer Experiments

In this section we present the results of our computer experiments with the direct attack against HMFEv schemes over small fields. In particular, we wanted to answer the questions

1. Is the concrete choice of k and v (or only the sum) important for the degree of regularity of a direct attack against the scheme? and
2. Is the upper bound on d_{reg} given by equation (4) reasonable tight?

In order to answer the first question, we performed experiments of the following type: For fixed values of q and $s = k + v$, we varied the values of k and v. We then created the public systems of the corresponding HMFEv instances (for different values of ℓ) and solved these systems using the F_4 algorithm integrated in MAGMA. The experiments were (like all the experiments presented in this paper) performed on a server with 16 AMD Opteron cores (2.4 GHz) and 128 GB of RAM. However, as MAGMA is not parallelizable, our programs use only one core.

In our experiments, we fixed the field $\mathbb{F}$ to be GF(2) and the sum $s = k+v$ to be 9. We varied v in the interval $I = \{0, \ldots, 8\}$ and created HMFEv(GF(2),$s - v$, ℓ, v) instances (for increasing values of ℓ). After that, we fixed v of the variables to get a determined system and solved the resulting public systems by the F_4 algorithm integrated in MAGMA. Table 4 shows, for $v \in I$, the highest degree of regularity we observed in these experiments. For each parameter set, we performed 10 experiments.

As the experiments show, the concrete ratio between k and v has, as long as we

v	0	1	2	3	4	5	6	7	8
k	9	8	7	6	5	4	3	2	1
d_{reg}	3	4	4	5	5	5	5	5	4

Table 4. Degree of regularity of HMFEv systems over GF(2) with $k + v = 9$

choose v and k not too small, no influence on the degree of regularity of solving the public systems of HMFEv. For HMFEv schemes over larger fields the importance of the concrete choice of k and v decreases further, since those systems

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

behave much more like random systems (see Section 6). We therefore choose, in order to increase the efficiency of our scheme, the parameter $k \in \{2, 3\}$ and increase v to reach the required level of security.

Is the upper bound on $d_{\mathbf{reg}}$ given by equation (4) reasonable tight?

In this section we want to analyze the question whether the upper bound on the degree of regularity given by equation (4) is reasonable tight. To do this, we created for fixed values of q, k and v and varying values of ℓ public systems of HMFEv and solved them with the F_4 algorithm integrated in MAGMA. We increased the value of ℓ and therefore the numbers of equations and variables in the system until we reached the upper bound of (4) or ran out of memory.
It is obvious that we can only hope to find such systems for small field sizes. We therefore restricted to values of $q \in \{2, 3\}$.

By doing so, we identified the following "tight" instances of HMFEv

scheme	upper bound on d_{reg} (equation (4))	experimental result
HMFEv-(GF(2),1,ℓ,2)	3	3 for $\ell \geq 9$ ($n \geq 9$)
HMFEv-(GF(2),2,ℓ,3)	4	4 for $\ell \geq 9$ ($n \geq 18$)
HMFEv-(GF(2),3,ℓ,4)	5	5 for $\ell \geq 10$ ($n \geq 30$)
HMFEv-(GF(3),1,ℓ,2)	5	5 for $\ell \geq 18$ ($n \geq 18$)

For most other HMFEv instances with $q \in \{2, 3\}$ and $k + v \leq 9$ we missed the upper bound given by equation (4) only by 1.
We believe that, also for these systems, we could have reached the upper bound given by equation (4) by increasing the parameter ℓ further. However, we did not have the necessary memory resources to solve HMFEv systems with more than 35 equations.

B Efficient Implementation of the Public and Private Maps of HMFEv

The most costly step during the signature generation process of our scheme is the inversion of the central equation $\mathcal{F}_V(\mathbf{Y}) = \mathbf{X}$, which is given as a system of k multivariate quadratic equations in k variables over the extension field $\mathbb{E}$. Since the coefficients of this system are chosen randomly, we need a system solver like a XL or a Gröbner basis algorithm for this step.
Obviously, the complexity of solving the system $\mathcal{F}_V(\mathbf{Y}) = \mathbf{X}$ and therefore the complexity of the signature generation process depends mainly on the choice of the parameter k. A small value of k will reduce the number of $\mathbb{E}$-multiplications in this process. However, it also leads to large extension fields and therefore increases the cost of a single $\mathbb{E}$-multiplication. Furthermore, choosing k too small might weaken the security of our scheme (see Section 7.1).

To find the optimal parameter k for our scheme, we therefore have to analyze the

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

process of inverting the central map $\mathcal{F}_V$ in more detail. Let the multivariate system $\mathcal{F}_V$ be given by the k multivariate quadratic maps $f_V^{(1)}, \ldots, f_V^{(k)} : \mathbb{E}^k \to \mathbb{E}$. As we find, the process of solving the multivariate system $\mathcal{F}_V(\mathbf{Y}) = \mathbf{X}$ consists mainly of two parts:

1. (**Gröbner basis step**) Find a univariate polynomial $p : \mathbb{E} \to \mathbb{E}$ in the ideal $\langle f_V^{(1)}, \ldots, f_V^{(k)} \rangle$.
2. (**Solving Step**) Solve the polynomial p by Berlekamp's algorithm.

In the following we analyze, for different values of k, these two steps in detail. For this, we fix the number $n = k \cdot \ell$ to $n = 48$ and choose $k \in \{2, 3, 4\}$. Inverting the system $\mathcal{F}_V$ therefore relates to

- solving a system of 2 quadratic equations in 2 variables over $\mathbb{F}^{24}$ or
- solving a system of 3 quadratic equations in 3 variables over $\mathbb{F}^{16}$ or
- solving a system of 4 quadratic equations in 4 variables over $\mathbb{F}^{12}$.

For $k = 2, 3$, we use for the first part a specially designed Gröbner basis method tailored for the occasion. In the case of 2 quadratic equations in 2 variables, we run in the Gröbner basis step successively 2 Gaussian eliminations on matrices of size 5×9 and 7×10. By doing so, we obtain a single variable equation p of degree 4. To perform this step, we need about $5 \cdot (11 + 12) + 7 \cdot 8 \cdot 4 = 339$ multiplications over the field $\mathbb{F}^{24}$.
In the Solving step, we have to solve the univariate equation p of degree 4 over the field $\mathbb{F}^{24}$. This takes about $6 \cdot 4^2 \cdot 24 = 2,304$ multiplications over the field of size $\mathbb{F}^{24}$. One can see that the overall complexity is dominated by the Solving step.

In the case of 3 quadratic equations in 3 variables, we run in the Gröbner basis step successively 3 Gaussian eliminations on matrices of size 11×19, 8×16 and 5×13 with many zero elements to derive a single variable equation of degree 8. For this we need about 1,700 $\mathbb{F}_{16}$ multiplications.
Then we solve this single variable equation of degree 8 over the larger field. This requires about $6 \cdot 8^2 \cdot 16 = 6,144$ big-field multiplications. One can see that the Solving step dominates the complexity.

In the case of 4 quadratic equations in 4 variables, the situation is too complicated to do it by hand and we use the F_4 algorithm directly. In this case, we run successively Gaussian eliminations on matrices of size $19 \times 34, 41 \times 50, 42 \times 50$ and 35×48, which requires about $2 \cdot 50^3 = 250,000$ $\mathbb{F}^{12}$ multiplications. By doing so, we obtain a single variable equation p of degree 16.
In the Solving Step, we have to solve this univariate equation p over the larger field, which requires about $6 \cdot 16^2 \cdot 12 = 18,432$ multiplications. One can see that here the solving of the single variable equation does not dominate the complexity anymore.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

C Arithmetic in the Public and Private Maps of HMFEv

Evaluating the public map requires first to generate all monomials, and then the computation of the inner product polynomials from known monomials. The first step requires $n(n+1)/2$ field multiplications. The second part is much more important and requires $mn(n+3)/2$ multiplications in the field and nearly as many additions (or XORs) to accumulate the results.
Arithmetic in GF(256) is done via the table-lookup instruction VPSHUFB. This instruction allows 32 simultaneous lookups from a table of 16, which allows for easy scalar-vector multiplications of GF(16) using log-exp tables. Every 32 GF(16) multiplications then take two VPSHUFB instructions and an add in addition to the required VPXOR, because we store the public key in log form. Finally we put together multiplications of GF(256) for the public key using four multiplications in GF(16) (schoolbook method).
The main computation in big binary fields uses PCLMULQDQ and schoolbook because on recent processors this instruction is really fast. We also use lazy reductions,which means that we often do not reduce to the lowest degree. A time-constant complete reduction is performed after the entire operation.

Arithmetic in GF(31) uses AVX2 instructions (and following that SSSE3 instructions). For best use of our resources, we use a YMM register to represent a vector of 16 or 32 coefficients in the public key to be multiplied by two monomials. Values for two monomials each time are also expanded into an YMM register. The actual arithmetic uses the VPMADDUSBW instruction to multiply two pairs of byte values (one signed one unsigned) into signed 16-bit values, and add them together all in one cycle. This requires us to ensure that input monomials are in $0, \ldots, 31$ and the coefficients in $-15, \ldots, 15$. We add together 32 results of VPMADDUSBW each time, which keeps the result between ± 32767. We can then reduce the results again to between $0, \ldots, 31$.
Arithmetic in tower fields over GF(31) are in straight schoolbook form and do not use the VPMADDUSBW instruction because the sizes are not convenient for it.

Chen, Ming-Shing; Ding, Jintai; Petzoldt, Albrecht; Yang, Bo-Yin.
"HMFEv - An Efficient Multivariate Signature Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme

Dustin Moody[1], Ray Perlner[1], and Daniel Smith-Tone[1,2]

[1]National Institute of Standards and Technology,
Gaithersburg, Maryland, USA
[2]Department of Mathematics, University of Louisville,
Louisville, Kentucky, USA

dustin.moody@nist.gov, ray.perlner@nist.gov, daniel.smith@nist.gov

Abstract. In the last few years multivariate public key cryptography has experienced an infusion of new ideas for encryption. Among these new strategies is the ABC Simple Matrix family of encryption schemes which utilize the structure of a large matrix algebra to construct effectively invertible systems of nonlinear equations hidden by an isomorphism of polynomials. One promising approach to cryptanalyzing these schemes has been structural cryptanalysis, based on applying a strategy similar to MinRank attacks to the discrete differential. These attacks however have been significantly more expensive when applied to parameters using fields of characteristic 2, which have been the most common choice for published parameters. This disparity is especially great for the cubic version of the Simple Matrix Encryption Scheme.

In this work, we demonstrate a technique that can be used to implement a structural attack which is as efficient against parameters of characteristic 2 as are attacks against analogous parameters over higher characteristic fields. This attack demonstrates that, not only is the cubic simple matrix scheme susceptible to structural attacks, but that the published parameters claiming 80 bits of security are less secure than claimed (albeit only slightly.) Similar techniques can also be applied to improve structural attacks against the original Simple Matrix Encryption scheme, but they represent only a modest improvement over previous structural attacks. This work therefore demonstrates that choosing a field of characteristic 2 for the Simple Matrix Encryption Scheme or its cubic variant will not provide any additional security value.

Key words: multivariate public key cryptography, differential invariant, MinRank, encryption

1 Introduction

The National Institute of Standards and Technology (NIST) is currently engaged in an effort to update the public key infrastructure, providing alternatives to the classical public key schemes based on arithmetic constructions. The discovery by Peter Shor in the 1990s of efficient algorithms for factoring and computing discrete logarithms, see [1], accelerated research towards building the necessary class of computers, those that Feynman famously suggested in [2]: quantum computers. There has been growing interest among scientists in our discipline in the years since, to provide protocols and algorithms that are post-quantum, that is, secure in the quantum model of computing. The recent publication by (NIST), see [3], of a call for proposals for post-quantum standards directly addresses the challenge of migration towards a more diverse collection of tools for our public key infrastructure.

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Public key schemes based on the difficulty of inverting nonlinear systems of equations provide one possibility for post-quantum security. Multivariate Public Key Cryptography (MPKC) is a reasonable option because the problem of solving systems of nonlinear equations, even if only quadratic, is known to be NP-complete; thus, the generic problem is likely beyond the reach of quantum adversaries. Furthermore, there are a variety of standard techniques to metamorphosize multivariate schemes, to introduce new properties, to enhance security, to reduce power consumption, to resist side-channel analysis, etc.

There are numerous long-lived multivariate digital signature schemes. All of UOV [4], HFE- [5], and HFEv- [6] have been studied for around two decades. Moreover, some of the above schemes have optimizations which have strong theoretical support or have stood unbroken in the literature for some time. Notable among these are UOV, which has a cyclic variant [7] that dramatically reduces the key size, and Gui [8], an HFEv- scheme, that, due to tighter bounds on the complexity of algebraically solving the underlying system of equations, see [9], has much more aggressive parameters than QUARTZ, see [6].

Multivariate public key encryption, however, has a much rockier history. Several attempts at multivariate encryption, see [10, 11] for example, have been shown to be weak based on rank or differential weaknesses. Recently, a new framework for developing secure multivariate encryption schemes has surfaces, drawing on the idea that it may impose sufficiently few restrictions on a multivariate map to be merely an injective map into a much larger codomain instead of being essentially a permutation. A few interesting attempts to achieve multivariate encryption have originated from this thought. ZHFE, see [12], the quadratic and cubic variants of the ABC Simple Matrix Scheme, see [13] and [14], and Extension Field Cancellation, see [15], all use fundamentally new structures for the derivation of an encryption system.

A few of the above schemes have already suffered some setbacks. A questionable rank property in the public key of ZHFE presented in [16] makes this scheme appear dubious, while it was shown that the quadratic Simple Matrix structure leaves the signature of a differential invariant in the public key which is exploited in [17] to effect an attack.

The case of the Cubic Simple Matrix encryption scheme is more interesting; the authors in [14] present a heuristic argument for security and suggest the possibility of provable security for the scheme. These provable security claims were undermined in [18], however, with the presentation of a key recovery attack on a full scale version of the Cubic Simple Matrix encryption scheme. The complexity of the attack was on the order of q^{s+2} for characteristic $p > 3$, q^{s+3} for characteristic 3, and q^{2s+6} for characteristic 2. Here s is the dimension of the matrices in the scheme, and q is the cardinality of the finite field used. This technique was an extension and augmentation of the technique of [17], and similarly exploited a differential invariant property of the core map to perform a key recovery attack. Nonetheless, the much higher complexity of this attack for characteristic 2 left open the possibility that there may be some security advantage to using a cubic ABC map over a field with characteristic 2.

In this paper, we present an attack whose complexity is on the order of q^{s+2} for all characteristics. Similar techniques can also improve the complexity of attacks against characteristic 2 parameters for the original quadratic version of the ABC cryptosystem, from q^{s+4} (reported in [17]) to q^{s+2}.

Specifically, our technique improves the complexity of attacking CubicABC($q = 2^8$,$s = 7$), designed for 80-bit security, from the horrendous value of 2^{177} in [18] to approximately 2^{88} operations, the same as the direct algebraic attack complexity reported in [14]. More convincing is our attack on CubicABC($q = 2^8$,$s = 8$), designed for 100-bit security. We break the scheme in approximately 2^{98} operations. Furthermore, the attack is fully parallelizable and requires very little memory; hence, our technique is asymptotically far more efficient than algebraic attacks, the basis for the original security estimation. Thus, the security claims in [14] not only fail to hold in the odd characteristic case, they fail to hold in characteristic two as well.

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

The paper is organized as follows. In the next section, we present the structure of the Cubic ABC Simple Matrix encryption scheme. In the following section, the fingerprint of the matrix algebra used in the construction of the ABC scheme is exposed. In the subsequent section, the effect of this structure on minrank calculations is determined. We then calculate the complexity of the full attack including the linear algebra steps required for full key recovery. Finally, we review these results and discuss the security of the Cubic ABC scheme and its quadratic counterpart moving forward.

2 The Cubic ABC Matrix Encryption Scheme

In [14], the Cubic ABC Matrix encryption scheme is proposed. The motivation behind the scheme is to use a large matrix algebra over a finite field to construct an easily invertible cubic map. The construction uses matrix multiplication to combine random linear and quadratic formulae into cubic formulae in a way that allows a user with knowledge of the structure of the matrix algebra and the polynomial isomorphism used to compose the scheme to invert the map.

Let $k = \mathbb{F}_q$ be a finite field. Linear forms and variables over k will be denoted with lower case letters. Vectors of any dimension over k will be denoted with bold font, $\mathbf{v}$. Fix $s \in \mathbb{N}$ and set $n = s^2$ and $m = 2s^2$. An element of a matrix ring $M_d(k)$ or the linear transformations they represent, will be denoted by upper case letters, such as M. When the entries of the matrix are being considered functions of a variable, the matrix will be denoted $M(\mathbf{x})$. Let $\phi : M_{s\times 2s}(k) \to k^{2s^2}$ represent the vector space isomorphism sending a matrix to the column vector consisting of the concatenation of its rows. The output of this map, being a vector, will be written with bold font; however, to indicate the relationship to its matrix preimage, it will be denoted with an upper case letter, such as $\mathbf{M}$.

The scheme utilizes an isomorphism of polynomials to hide the internal structure. Let $\mathbf{x} = \left[x_1, x_2, \ldots, x_n\right]^\top \in k^n$ denote plaintext while $\mathbf{y} = \left[y_1, \ldots, y_m\right] \in k^m$ denotes ciphertext. Fix two invertible linear transformations $T \in M_m(k)$ and $U \in M_n(k)$. (One may use affine transformations, but there is no security or performance benefit in doing so.) Denote the input and output of the central map by $\mathbf{u} = U\mathbf{x}$ and $\mathbf{v} = T^{-1}(\mathbf{y})$.

The construction of the central map is as follows. Define three $s \times s$ matrices A, B, and C in the following way:

$$A = \begin{bmatrix} p_1 & p_2 & \cdots & p_s \\ p_{s+1} & p_{s+2} & \cdots & p_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ p_{s^2-s+1} & p_{s^2-s+2} & \cdots & p_{s^2} \end{bmatrix}, B = \begin{bmatrix} b_1 & b_2 & \cdots & b_s \\ b_{s+1} & b_{s+2} & \cdots & b_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ b_{s^2-s+1} & b_{s^2-s+2} & \cdots & b_{s^2} \end{bmatrix},$$

and

$$C = \begin{bmatrix} c_1 & c_2 & \cdots & c_s \\ c_{s+1} & c_{s+2} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{s^2-s+1} & c_{s^2-s+2} & \cdots & c_{s^2} \end{bmatrix}.$$

Here the p_i are quadratic forms on $\mathbf{u}$ chosen independently and uniformly at random from among all quadratic forms and the b_i and c_i are linear forms on $\mathbf{u}$ chosen independently and uniformly at random from among all linear forms.

We define two $s \times s$ matrices $E_1 = AB$ and $E_2 = AC$. Since A is quadratic and B and C are linear in u_i, E_1 and E_2 are cubic in the u_i. The central map $\mathcal{E}$ is defined by

$$\mathcal{E} = \phi \circ (E_1 || E_2).$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

Thus $\mathcal{E}$ is an m dimensional vector of cubic forms in $\mathbf{u}$. Finally, the public key is given by $\mathcal{F} = T \circ \mathcal{E} \circ U$.

Encryption with this system is standard: given a plaintext $(x_1, \ldots, x_n)$, compute $(y_1, \ldots, y_m) = \mathcal{F}(x_1, \ldots, x_n)$. Decryption is somewhat more complicated.

To decrypt, one inverts each of the private maps in turn: apply T^{-1}, invert $\mathcal{E}$, and apply U^{-1}. To "invert" $\mathcal{E}$, one assumes that $A(\mathbf{u})$ is invertible, and forms a matrix

$$A^{-1}(\mathbf{u}) = \begin{bmatrix} w_1 & w_2 & \cdots & w_s \\ w_{s+1} & w_{s+2} & \cdots & w_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ w_{s^2-s+1} & w_{s^2-s+2} & \cdots & w_{s^2} \end{bmatrix},$$

where the w_i are indeterminants. Then collectinging the relations $A^{-1}(\mathbf{u})E_1(\mathbf{u}) = B(\mathbf{u})$ and $A^{-1}(\mathbf{u})E_2(\mathbf{u}) = C(\mathbf{u})$, we have $m = 2s^2$ linear equations in $2n = 2s^2$ unknowns w_i and u_i. Using, for example, Gaussian elimination one can eliminate all of the variables w_i and most of the u_i. The resulting relations can be substituted back into $E_1(\mathbf{u})$ and $E_2(\mathbf{u})$ to obtain a large system of equations in very few variables which can be solved efficiently in a variety of ways.

3 The Structure of the Cubic ABC scheme

3.1 Column Band Spaces

Each component of the central $\mathcal{E}(\mathbf{u}) = E_1(\mathbf{u})||E_2(\mathbf{u})$ map may be written as:

$$\mathcal{E}_{(i-1)s+j} = \sum_{l=1}^{s} p_{(i-1)s+l} b_{(l-1)s+j},$$

for the E_1 equations, and likewise, for the E_2 equations:

$$\mathcal{E}_{s^2+(i-1)s+j} = \sum_{l=1}^{s} p_{(i-1)s+l} c_{(l-1)s+j}$$

where i and j run from 1 to s.

Consider the s sets of s polynomials that form the columns of E_1, i.e. for each $j \in \{1, \ldots, s\}$ consider $(\mathcal{E}_j, \mathcal{E}_{s+j}, \ldots, \mathcal{E}_{s^2-s+j})$. With high probability, the linear forms $b_j, b_{s+j}, \ldots, b_{s^2-s+j}$ are linearly independent, and if so the polynomials may be re-expressed, using a linear change of variables to $(u'_1, \ldots u'_{s^2})$ where $u'_i = b_{(i-1)s+j}$ for $i = 1, \ldots, s$. After the change of variables, the only cubic monomials contained in $(\mathcal{E}_j, \mathcal{E}_{s+j}, \ldots, \mathcal{E}_{s^2-s+j})$ will be those containing at least one factor of $u'_1, \ldots, u'_s$. We can make a similar change of variables to reveal structure in the s sets of s polynomials that form the columns of E_2: Setting $u'_i = c_{(i-1)s+j}$ for $i = 1, \ldots, s$ and a fixed j, the only cubic monomials contained in $(\mathcal{E}_{s^2+j}, \mathcal{E}_{s^2+s+j}, \ldots, \mathcal{E}_{2s^2-s+j})$ will be those containing at least one factor of $u'_1, \ldots, u'_s$.

More generally, we can make a similar change of variables to reveal structure in any of a large family of s dimensional subspaces of the span of the component polynomials of E_1 and E_2, which we will call column band spaces in analogy to the band spaces used to analyze the quadratic ABC cryptosystem in [17]. Each family is defined by a fixed linear combination, (β, γ), of the columns of E_1 and E_2:

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

Definition 1 *The column band space defined by the $2s$-dimensional linear form (β, γ) is the space of cubic maps, $\mathcal{B}_{\beta,\gamma}$, given by:*

$$\mathcal{B}_{\beta,\gamma} = Span(\mathcal{E}_{\beta,\gamma,1}, \ldots, \mathcal{E}_{\beta,\gamma,s}),$$

where

$$\mathcal{E}_{\beta,\gamma,i} = \sum_{j=1}^{s} (\beta_j \mathcal{E}_{(i-1)s+j} + \gamma_j \mathcal{E}_{s^2+(i-1)s+j})$$

$$= \sum_{l=1}^{s} \left(p_{(i-1)s+l} \sum_{j=1}^{s} \left(\beta_j b_{(l-1)s+j} + \gamma_j c_{(l-1)s+j} \right) \right).$$

Note that under a change of variables

$$(x_1, \ldots, x_{s^2}) \xmapsto{M} (u'_1, \ldots u'_{s^2}), \text{ where } u'_i = \sum_{j=1}^{s} \left(\beta_j b_{(i-1)s+j} + \gamma_j c_{(i-1)s+j} \right) \text{ for } i = 1, \ldots, s,$$

the only cubic monomials contained in the elements of $\mathcal{B}_{\beta,\gamma}$ will be those containing at least one factor of $u'_1, \ldots, u'_s$.

In such a basis, the third formal derivative, or the 3-tensor of third partial derivatives

$$D^3\mathcal{E} = \sum_{i,j,k} \frac{\partial^3 \mathcal{E}}{\partial u'_i \partial u'_j \partial u'_k} du'_i \otimes du'_j \otimes du'_k,$$

of any map $\mathcal{E} \in \mathcal{B}_{\beta,\gamma}$ has a special block form, see Figure 1. This tensor is the same as the one used for the attack in [18], although in that case it was computed using the discrete differential. There are, however, a number of disadvantages to using this 3-tensor to represent the structural features of cubic ABC. In particular, when defined over a field of characteristic 2, the symmetry of the 3-tensor results in the loss of any information about coefficients for monomials of the form $x_i^2 x_j$, since the 3rd derivatave of such a monomial is always 0. We will therefore use a different tool to express the structure of cubic ABC.

Using the same u' basis as above, we see that the gradient $\nabla_{u'}\mathcal{E}$ produces a covector of quadratic forms, which can be though of as a quadratic map that takes any vector w of the form

$$(0, \ldots, 0, u'_{s+1}(\mathbf{w}), \ldots, u'_{s^2}(\mathbf{w}))^\top,$$

to a covector of the form

$$(y(u'_1), \ldots, y(u'_s), 0, \ldots, 0).$$

Note that, by the chain rule, we can relate $\nabla_{u'}\mathcal{E} = \left[\frac{\partial \mathcal{E}}{\partial u'_1}, \ldots, \frac{\partial \mathcal{E}}{\partial u'_{s^2}} \right]$ to the formal derivative defined over the public basis:

$$\nabla \mathcal{E} = \left[\frac{\partial \mathcal{E}}{\partial x_1}, \ldots, \frac{\partial \mathcal{E}}{\partial x_{s^2}} \right] = \nabla_{u'}\mathcal{E} \left[\frac{du'_j}{dx_i} \right]_{i,j}$$

using the nonsingular change of basis matrix whose entries are $\frac{du'_j}{dx_i}$. We can therefore conclude that even defined over the public basis, the first formal derivative of any map $\mathcal{E} \in \mathcal{B}_{\beta,\gamma}$ is a quadratic map that takes an $s^2 - s$ dimensional space of vectors to an s dimensional space of covectors.

We will define the term "band kernel" to describe this $s^2 - s$ dimensional space of vectors (including $\mathbf{w}$) which are mapped to an s dimensonal image space by the first formal derivative of $\mathcal{E}$.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

Definition 2 *The band kernel of $\mathcal{B}_{\beta,\gamma}$, denoted $\mathcal{BK}_{\beta,\gamma}$, is the space of vectors x, such that*

$$u'_i = \sum_{j=1}^{s} \beta_j b_{(i-1)s+j}(x) + \gamma_j c_{(i-1)s+j}(x) \ = 0,$$

for $i = 1, \ldots, s$.

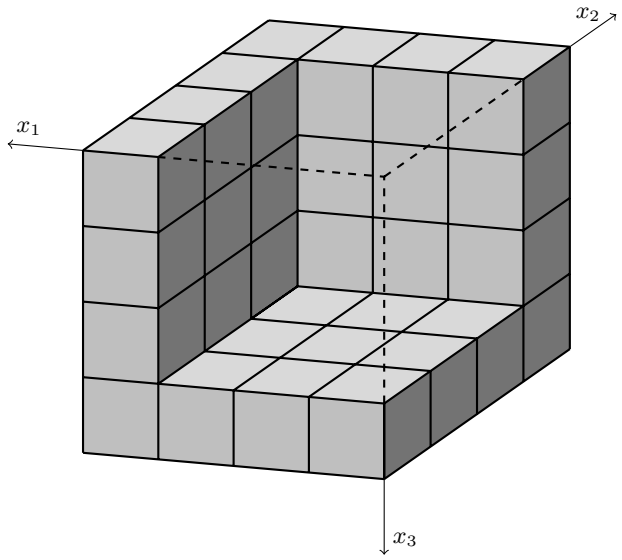


Fig. 1. 3-tensor structure of the third formal derivative of a band space map. Solid regions correspond to nonzero coefficients. Transparent regions correspond to zero coefficients.

4 A Variant of MinRank Exploiting the Column Band Space Structure

A minrank-like attack may be used to locate the column band space maps defined in the previous section. In this case, the attack proceeds by selecting s^2-dimensional vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, setting

$$\begin{aligned} \sum_{i=1}^{2s^2} t_i \nabla \mathcal{E}_i(\mathbf{w}_1) = 0, \\ \sum_{i=1}^{2s^2} t_i \nabla \mathcal{E}_i(\mathbf{w}_2) = 0, \end{aligned} \tag{1}$$

and then solving for the t_i. The attack succeeds when $\sum_{i=1}^{2s^2} t_i \mathcal{E}_i \in \mathcal{B}_{\beta,\gamma}$, and $\mathbf{x}_1$ and $\mathbf{x}_2$ are within the corresponding band kernel. If these conditions are met, then the 2-tensors

$$\sum_{i=1}^{2s^2} t_i \mathbf{H}(\mathcal{E}_i)(\mathbf{w}_1) \text{ and } \sum_{i=1}^{2s^2} t_i \mathbf{H}(\mathcal{E}_i)(\mathbf{w}_2),$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

will have rank at most $2s$, and this will be easily detectable. Here $\mathbf{H}(\mathcal{E}_i)$ is the Hessian matrix

$$\mathbf{H}(\mathcal{E}_i) := \begin{bmatrix} \frac{\partial^2 \mathcal{E}_i}{\partial x_1^2} & \frac{\partial^2 \mathcal{E}_i}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \mathcal{E}_i}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \mathcal{E}_i}{\partial x_1 \partial x_2} & \frac{\partial^2 \mathcal{E}_i}{\partial x_2^2} & \cdots & \frac{\partial^2 \mathcal{E}_i}{\partial x_1 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathcal{E}_i}{\partial x_n \partial x_1} & \frac{\partial^2 \mathcal{E}_i}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \mathcal{E}_i}{\partial x_n^2} \end{bmatrix}.$$

Theorem 1 *The probability that 2 randomly chosen vectors, $\mathbf{w}_1$ and $\mathbf{w}_2$, are both in the band kernel of some band space $\mathcal{B}_{\beta,\gamma}$ is approximately $\frac{1}{q-1}$.*

Proof. The condition that the $\mathbf{w}_1$ and $\mathbf{w}_2$ are contained within a band kernel is that there be a nontrivial linear combination of the columns of the following matrix which is equal to zero (i.e. that the matrix has nonzero column corank):

$$\left[\begin{array}{cccc|cccc} b_1(\mathbf{w}_1) & b_2(\mathbf{w}_1) & \ldots & b_s(\mathbf{w}_1) & c_1(\mathbf{w}_1) & c_2(\mathbf{w}_1) & \ldots & c_s(\mathbf{w}_1) \\ b_{s+1}(\mathbf{w}_1) & b_{s+2}(\mathbf{w}_1) & \ldots & b_{2s}(\mathbf{w}_1) & c_{s+1}(\mathbf{w}_1) & c_{s+2}(\mathbf{w}_1) & \ldots & c_{2s}(\mathbf{w}_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{s^2-s+1}(\mathbf{w}_1) & b_{s^2-s+2}(\mathbf{w}_1) & \ldots & b_{s^2}(\mathbf{w}_1) & c_{s^2-s+1}(\mathbf{w}_1) & c_{s^2-s+2}(\mathbf{w}_1) & \ldots & c_{s^2}(\mathbf{w}_1) \\ \hline b_1(\mathbf{w}_2) & b_2(\mathbf{w}_2) & \ldots & b_s(\mathbf{w}_2) & c_1(\mathbf{w}_2) & c_2(\mathbf{w}_2) & \ldots & c_s(\mathbf{w}_2) \\ b_{s+1}(\mathbf{w}_2) & b_{s+2}(\mathbf{w}_2) & \ldots & b_{2s}(\mathbf{w}_2) & c_{s+1}(\mathbf{w}_2) & c_{s+2}(\mathbf{w}_2) & \ldots & c_{2s}(\mathbf{w}_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{s^2-s+1}(\mathbf{w}_2) & b_{s^2-s+2}(\mathbf{w}_2) & \ldots & b_{s^2}(\mathbf{w}_2) & c_{s^2-s+1}(\mathbf{w}_2) & c_{s^2-s+2}(\mathbf{w}_2) & \ldots & c_{s^2}(\mathbf{w}_2) \end{array}\right].$$

The matrix is a uniformly random $2s \times 2s$ matrix, which has nonzero column corank with probability approximately $\frac{1}{q-1}$. □

Theorem 2 *If $\mathbf{w}_1$ and $\mathbf{w}_2$ are chosen in such a way that they are both in the band kernel of a column band space $\mathcal{B}_{\beta,\gamma}$, and they are linearly independent from one another and statistically independent from the private quadratic forms, $p_{(i-1)s+j}$ in the matrix A, then $\mathbf{w}_1$ and $\mathbf{w}_2$ are both in the kernel of the first formal derivative of some column band space map, $\mathcal{E} = \sum_{\mathcal{E}_{\beta,\gamma,i} \in \mathcal{B}_{\beta,\gamma}} \tau_i \mathcal{E}_{\beta,\gamma,i}$ with probability approximately $\frac{1}{(q-1)q^s}$.*

Proof. An $\mathcal{E}$ meeting the above condition exists iff there is a nontrivial solution to the following system of equations

$$\begin{aligned} \sum_{\mathcal{E}_{\beta,\gamma,i} \in \mathcal{B}_{\beta,\gamma}} \tau_i \nabla \mathcal{E}_{\beta,\gamma,i}(\mathbf{w}_1) = 0, \\ \sum_{\mathcal{E}_{\beta,\gamma,i} \in \mathcal{B}_{\beta,\gamma}} \tau_i \nabla \mathcal{E}_{\beta,\gamma,i}(\mathbf{w}_2) = 0. \end{aligned} \tag{2}$$

We may express our band space maps in a basis (e.g. the u'_i basis used in Definition 2) where the first s basis vectors are chosen to be outside the band kernel, and the remaining $s^2 - s$ basis vectors are chosen from within the band kernel. Combining this with Definition 1, we see that the band space maps can be written as

$$\mathcal{E}_{\beta,\gamma,i} = \sum_{j=1}^{s} p_{(i-1)s+j} u'_j.$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

Note that $\mathbf{w}_1$ and $\mathbf{w}_2$ are band kernel vectors, and so for both vectors we have that $u'_j = 0$ for $j = 1, \ldots, s$. Therefore, in such a basis, the only formal derivatives of $\mathcal{E}$ that can be nonzero are $\frac{\partial \mathcal{E}}{\partial u'_j} = p_{(i-1)s+j}$ for $j = 1, \ldots, s$. Thus in order for there to be a nontrivial solution to Equation (2), it is necessary and sufficient that $\sum_{i=1}^{s} \tau_i p_{(i-1)s+j}(\mathbf{w}_k) = 0$ for $j = 1, \ldots, s$ and $k = 1, 2$. This condition will be satisfied if and only if the following $2s \times s$ matrix has nonzero column corank:

$$\left[\begin{array}{cccc} p_1(\mathbf{w}_1) & p_{s+1}(\mathbf{w}_1) & \cdots & p_{s^2-s+1}(\mathbf{w}_1) \\ p_2(\mathbf{w}_1) & p_{s+2}(\mathbf{w}_1) & \cdots & p_{s^2-s+2}(\mathbf{w}_1) \\ \vdots & \vdots & \ddots & \vdots \\ p_s(\mathbf{w}_1) & p_{2s}(\mathbf{w}_1) & \cdots & p_{s^2}(\mathbf{w}_1). \\ \hline p_1(\mathbf{w}_2) & p_{s+1}(\mathbf{w}_2) & \cdots & p_{s^2-s+1}(\mathbf{w}_2) \\ p_2(\mathbf{w}_2) & p_{s+2}(\mathbf{w}_2) & \cdots & p_{s^2-s+2}(\mathbf{w}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_s(\mathbf{w}_2) & p_{2s}(\mathbf{w}_2) & \cdots & p_{s^2}(\mathbf{w}_2) \end{array}\right].$$

This matrix is a random matrix over $k = \mathbb{F}_q$, which has nonzero column corank with probability approximately $\frac{1}{(q-1)q^s}$, for practical parameters. □

Combining the results of Theorems 1 and 2, we find that for a random choice of the vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, there is a column band space map among the solutions of Equation (1) with probability approximately $\frac{1}{(q-1)^2 q^s}$. It may be somewhat undesirable to choose $\mathbf{w}_1$ and $\mathbf{w}_1$ completely randomly, however. The naïve algorithm for constructing the coefficients of Equation (1) for a random choice of $\mathbf{w}_1$ and $\mathbf{w}_2$ requires on the order of s^8 field operations. This can be reduced to s^6 operations if we make sure that each new choice of $\mathbf{w}_1$ and $\mathbf{w}_2$ differs from the previous choice at only a single coordinate. Then, rather than recomputing Equation (1) from scratch, we can use the previous values of the coefficients and we will only need to include corrections for the monomials that contain the variable that was changed from the previous iteration. Over a large number of iterations, the distribution of $\mathbf{w}_1$ and $\mathbf{w}_2$ should still be sufficiently close to random that the probability of success for the attack will not be meaningfully altered.

One final factor which may increase the cost of attacks is the expected dimension of the solution space of Equation (1). If this space has a high dimension, then the attack will be slowed down since the attacker much search through a large number of spurious solutions to find a real solution (i.e. one where $\sum_{i=1}^{2s^2} t_i \mathbf{H}(\mathcal{E}_i)(\mathbf{w}_l)$ has rank at most $2s$ for $l = 1, 2$). Fortunately, Equation (1) is a system of $2s^2$ equations in $2s^2$ variables and it generally has a 0-dimensional space of solutions. The lone exception occurs for characteristic 3. In this case, there are two linear dependencies among the equations, given by $\mathbf{w}_1 \left[\nabla \mathcal{E}_i(\mathbf{w}_1)\right]^\top = 0$ and $\mathbf{w}_2 \left[\nabla \mathcal{E}_i(\mathbf{w}_2)\right]^\top = 0$. In this situation we would therefore expect a 2-dimensional solution space. We can, however, recover two additional linear constraints on the t_i's by also requiring:

$$\sum_{i=1}^{2s^2} t_i \mathcal{E}_i(\mathbf{w}_l) = 0, \text{ for } l = 1, 2.$$

When these additional linear constraints are added to those given by Equation (1), the expected dimension of the solution space drops back to 0. We can therefore assess the cost of the above attack at approximately $s^6 q^{s+2}$, regardless of the characteristic.

5 Application to the Quadratic ABC Scheme

A similar technique was used to attack the original quadratic version of the ABC cryptosytem in [17]. While this technique was expressed in terms of the discrete differential, it can also be

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

expressed using the formal derivative. In that case, the attack proceeds by selecting two random vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, and solving an equation identical to Equation (1) for t_i, where the $\mathcal{E}_i$ are quadratic rather than cubic. The attack succeeds when $\sum_{i=1}^{2s^2} t_i \mathbf{H}(\mathcal{E}_i)$ has low rank.

When this attack is applied to parameters chosen over a field with characteristic 2, it is less efficient for the same reason as the basic attack given in the previous section is less efficient for the characteristic 3 parameters: the $2s^2$ linear equations given by Equation (1) have three linear dependencies given by $\mathbf{w}_1 \left[\nabla \mathcal{E}_i(\mathbf{w}_1)\right]^\top = 0$, $\mathbf{w}_2 \left[\nabla \mathcal{E}_i(\mathbf{w}_2)\right]^\top = 0$, and $\mathbf{w}_1 \left[\nabla \mathcal{E}_i(\mathbf{w}_2)\right]^\top + \mathbf{w}_2 \left[\nabla \mathcal{E}_i(\mathbf{w}_1)\right]^\top = 0$, and the attacker must generally search through a 3-dimensional solution space of spurious solutions in order to find a 1-dimensional space of useful solutions. As a result, the complexity of the attack for characteristic 2 is $s^{2\omega} q^{s+4}$, instead of $s^{2\omega} q^{s+2}$, as it is for all other characteristics. ($\omega \approx 2.373$ is the linear algebra constant.)

However, just as with cubic ABC parameters of characteristic 3, we can add two additional linear constraints and reduce the expected dimension of the solution space to 1:

$$\sum_{i=1}^{2s^2} t_i \mathcal{E}_i(\mathbf{w}_l) = 0, \text{ for } l = 1, 2.$$

Thus, we can also reduce the attack complexity for quadratic ABC parameters with characteristic 2 to $s^{2\omega} q^{s+2}$.

6 Completing the Key Recovery

Once the MinRank instance is solved, key extraction proceeds in a similar manner to [18, Section 6] in the cubic case and [17, Section 6]. Here we discuss the cubic version.

First, note that U is not a critical element of the scheme. If A is a random matrix of quadratic forms and B and C are random matrices of linear forms, then so are $A \circ U$, $B \circ U$ and $C \circ U$ for any full rank map U. Thus, since $T \circ \phi(AB||AC) \circ U = T \circ \phi((A \circ U)(B \circ U)||(A \circ U)(C \circ U))$, we may absorb the action of U into A, B, and C, and consider the public key to be of the form

$$P(\mathbf{x}) = T \circ \phi(AB||AC)(\mathbf{x}).$$

Let $\mathcal{E} \in \mathcal{B}_{\beta,\gamma}$, and consider $\mathbf{H}(\mathcal{E})$. For $\mathbf{w}_1$ and $\mathbf{w}_2$ in the band kernel corresponding to $\mathcal{B}_{\beta,\gamma}$, there is a basis in which both $\mathbf{H}(\mathcal{E})(\mathbf{w}_1)$ and $\mathbf{H}(\mathcal{E})(\mathbf{w}_2)$ have the form illustrated in Figure 2. Thus, for $s \geq 3$, with high probability the kernels of both maps are contained in the corresponding band kernel $\mathcal{B}_{\beta,\gamma}$, and $\text{span}\{\ker(\mathbf{H}(\mathcal{E})(\mathbf{w}_1), \ker(\mathbf{H}(\mathcal{E})(\mathbf{w}_2)\} = \mathcal{B}_{\beta,\gamma}$.

Given the basis for an $s^2 - s$ dimensional band kernel $\mathcal{BK}$, we may choose a basis $\{v_1, \ldots, v_s\}$ for the subspace of the dual space vanishing on $\mathcal{BK}$. We can also find a basis $\mathcal{E}_{v_1}, \ldots, \mathcal{E}_{v_s}$ for the band space itself by solving the linear system

$$\begin{aligned}
\sum_{\mathcal{E}_i} \tau_i \mathcal{E}_i(\mathbf{w}_1) &= 0, \\
\sum_{\mathcal{E}_i} \tau_i \mathcal{E}_i(\mathbf{w}_2) &= 0, \\
\vdots &= \vdots \\
\sum_{\mathcal{E}_i} \tau_i \mathcal{E}_i(\mathbf{w}_t) &= 0,
\end{aligned}$$

where $t \approx 2s^2$ and $\mathbf{w}_i$ is in the band kernel.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

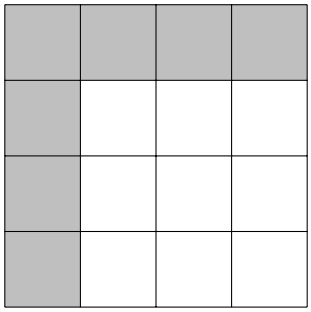


Fig. 2. Structure of $\mathbf{H}(\mathcal{E})(\mathbf{w})$ when $\mathcal{E} \in \mathcal{B}_{\beta,\gamma}$ and $\mathbf{w}$ is in the band kernel corresponding to the band space $\mathcal{B}_{\beta,\gamma}$. The shaded region corresponds to nonzero coefficients.

Since the basis $\mathcal{E}_{v_1}, \ldots, \mathcal{E}_{v_s}$ is in a single band space, there exists an element $\begin{bmatrix} b'_1 \cdots b'_s \end{bmatrix}^\top$ in ColumnSpace$(B||C)$, and two matrices Ω_1 and Ω_2 such that

$$\Omega_1 A \left(\Omega_2 \begin{bmatrix} b'_1 \\ \vdots \\ b'_s \end{bmatrix} \right) =: A' \left(\begin{bmatrix} v_1 \\ \vdots \\ v_s \end{bmatrix} \right) = \begin{bmatrix} \mathcal{E}_{v_1} \\ \vdots \\ \mathcal{E}_{v_s} \end{bmatrix}.$$

Solving the above system of equations over $\mathbb{F}_q[x_1, \ldots, x_{s^2}]$ uniquely determines A' in the quotient $\mathbb{F}_q[x_1, \ldots, x_{s^2}]/\langle v_1, \ldots, v_s \rangle$. To recover all of A', note that the above system is part of an equivalent key

$$\mathcal{F} = T' \circ A'(B'||C')$$

where $\begin{bmatrix} v_1 \cdots v_s \end{bmatrix}^\top$ is the first column of B'.

Applying T'^{-1} to both sides and inserting the information we know we may construct the system

$$A'(B'||C') = T'^{-1}\mathcal{F}. \tag{3}$$

Solving this system of equations modulo $\langle v_1, \ldots, v_s \rangle$ for B', C' and T'^{-1} we can recover a space of solutions, which we will restrict by arbitrarily fixing the value of T'^{-1}. Note that the elements of T'^{-1} are constant polynomials, and therefore $T'^{-1} (\mathrm{mod}\, \langle v_1, \ldots, v_s \rangle)$ is the same as T'^{-1}. Thus, for any choice of T'^{-1} in this space, the second column of $T'^{-1}\mathcal{F}$ is a basis for a band space. Moreover, the elements $v'_{s+1}, \ldots, v'_{2s}$ of the second column of $B' (\mathrm{mod}\, \langle v_1, \ldots, v_s \rangle)$ are the image, modulo $\langle v_1, \ldots, v_s \rangle$, of linear forms vanishing on the corresponding band kernel. Therefore, we obtain the equality

$$\left(\bigcap_{i=1}^{s} \ker(v_i) \right) \bigcap \left(\bigcap_{i=s+1}^{2s} \ker(v'_i) \right) = \mathcal{BK}_2 \cap \mathcal{BK}_1,$$

the intersection of the band kernels of our two band spaces.

We can reconstruct the full band kernel of this second band space using the same method we used to obtain our first band kernel. We take a map $\mathcal{E}_2$ from the second column of $T'^{-1}\mathcal{F}$, and two vectors $\mathbf{w}_a$ and $\mathbf{w}_b$ from $\mathcal{BK}_2 \cap \mathcal{BK}_1$, and we compute $\mathcal{BK}_2 = \mathrm{span}\{\ker(\mathbf{H}(\mathcal{E}_2)(\mathbf{w}_a) \cup \ker(\mathbf{H}(\mathcal{E}_2)(\mathbf{w}_b)\}$. We can now solve for the second column of B', $\begin{bmatrix} v_{s+1} \cdots v_{2s} \end{bmatrix}^\top$, uniquely over $\mathbb{F}_q[x_1, \ldots, x_{s^2}]$ (NOT modulo $\langle v_1, \ldots, v_s \rangle$) by solving the following system of linear equations:

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

$$\begin{aligned} v_i \equiv v_i' \bmod \langle v_1, \ldots, v_s \rangle, \\ v_i(\mathbf{w}_1) = 0, \\ v_i(\mathbf{w}_2) = 0, \\ \vdots = \vdots \\ v_i(\mathbf{w}_{s^2-s}) = 0, \end{aligned}$$

where $i = s+1, \ldots, 2s$, and $\{\mathbf{w}_1, \ldots, \mathbf{w}_{s^2-s}\}$ is a basis for $\mathcal{BK}_2$. We can now solve for A' (again, uniquely over $\mathbb{F}_q[x_1, \ldots, x_{s^2}]$) by solving:

$$A'\left(\begin{bmatrix} v_1 \\ \vdots \\ v_s \end{bmatrix}\right) \equiv \begin{bmatrix} \mathcal{E}_{v_1} \\ \vdots \\ \mathcal{E}_{v_s} \end{bmatrix} \bmod \langle v_1, \ldots, v_s \rangle,$$

$$A'\left(\begin{bmatrix} v_{s+1} \\ \vdots \\ v_{2s} \end{bmatrix}\right) \equiv \begin{bmatrix} \mathcal{E}_{v_{s+1}} \\ \vdots \\ \mathcal{E}_{v_{2s}} \end{bmatrix} \bmod \langle v_{s+1}, \ldots, v_{2s} \rangle,$$

where $\begin{bmatrix} \mathcal{E}_{v_{s+1}} & \cdots & \mathcal{E}_{v_{2s}} \end{bmatrix}^\top$ is the second column of $T'^{-1}\mathcal{F}$. This allows us to solve Equation (3) for the rest of B' and C', completing the attack.

The primary cost of the attack involves finding the band space map. The rest of the key recovery is additive in complexity and dominated by the band space map recovery; thus the total complexity of the attack is of the same order as the band space map recovery. Hence, the cost of private key extraction is approximately $q^{s+2}s^6$ for all characteristics.

The original parameters of Cubic ABC were designed for a security level of 80-bits and 100-bits. Since NIST has been recommending a security level of 112-bits since 2015, see [19], these figures may be a bit out of date. In fact, our attack seems more effective for larger parameter sets than small.

We note that our attack breaks CubicABC($q = 2^8, s = 7$), designed for 80-bit security, in approximately 2^{88} operations. More convincingly, our attack breaks CubicABC($q = 2^8, s = 8$), designed for 100-bit security, in approximately 2^{98} operations, indicating that for parameters as small as these, we have already crossed the threshold of algebraic attack efficiency. Furthermore, the attack is fully parallelizable and requires very little memory. Hence, this technique is asymptotically far more efficient than algebraic attacks, the basis for the original security estimation in [14].

In the case of the quadratic ABC scheme, the original 86-bit secure parameters ABC($q = 2^8, s = 8$). The attack complexity with the new methodology presented here is 2^{87}, just above the claimed level. We note, however, that the authors of [13] supplied additional parameters using odd characteristic in their presentation at PQCRYPTO 2013, see [20], with a claimed security level of 108-bits. This scheme, ABC($q = 127, s = 8$) offers resistance only to the level of 2^{77} to our slight improvement in technique over that of [17]. Thus, our attack definitively breaks these parameters.

7 Experiments

Using SAGE [21], we performed some experiments as a sanity check to confirm the efficiency of our ideas on small scale variants of the Cubic ABC scheme. The computer used has a 64 bit

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

quad-core Intel i7 processor, with clock cycle 2.8 GHz. Rather than considering the full attack, we were most interested in confirming our complexity estimates on the most costly step in the attack, the MinRank instance. Given as input the finite field size q, and the scheme parameter s, we computed the average number of vectors v required to be sampled in order for the rank of the 2-tensor $\mathbf{H}(\mathcal{E})(v)$ to fall to $2s$. As explained in Section 4, when the rank falls to this level, we have identified the subspace differential invariant structure of the scheme which can then be exploited to attack the scheme.

As this paper is only concerned with binary fields, we ran experiments with $q = 2, 4$ and 8. We found that for $s = 3$ and $q = 2, 4$, or 8, with high probability only a single vector was needed before the rank fell to $2s$. For $s = 4$ and $s = 5$, the computations were only feasible in SAGE for $q = 2$ and $q = 4$. The average values obtained are presented in the table below. Note that for $q = 4$ and $s = 5$ the average value is based on a small number of samples as the computation time was quite lengthy.

	$s=4$	$(q-1)^2q^s$	$s=5$	$(q-1)^2q^s$
$q=2$	24	16	35	32
$q=4$	1962	2304	7021	9216

Table 1. Average number of vectors needed for the rank to fall to $2s$ versus the predicted values.

In comparison, our previous experiments [18] were only able to obtain data for $q = 2$ and $s = 4, 5$. The average number of vectors needed in the $s = 4$ case was 244, while for $s = 5$, the average number in our experiments was 994 (with the predicted values being 256 and 1024).

8 Conclusion

The ABC schemes offer an interesting new technique for the construction of multivariate public key schemes. Previously, we have used the multiplicative structure of an extension field to generate an efficiently invertible map. Schemes built on such a construct are known as "big field" schemes. The ABC framework is essentially a "large structure" or perhaps "large algebra" scheme, depending on multiplication from a matrix algebra over the base field. Since the only simple algebras are either matrix algebras or field extensions, we seem to have exhausted the possibilities. Interestingly, MinRank techniques seem optimal in this setting, at least asymptotically in the dimension of the extension.

Also interesting to note is the fact that the authors present in [14] a heuristic security argument for the provable security of the scheme and reinforce the notion of provable security in this venue at the presentation of the scheme at [22]. Unfortunately, this analysis does not contribute a sound conclusion, as demonstrated by the methodology of [18]. With our improved attack, we rule out the possibility that the cubic variant of ABC offers any security advantage over the original quadratic scheme. Likewise, our improved attack on quadratic ABC eliminates any security benefit associated with characteristic-2 parameters in the quadratic case.

References

1. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Sci. Stat. Comp. **26, 1484** (1997)
2. Feynman, R.P.: Simulating physics with computers. Int. J. Theor. Phys. **21** (1982) 467–488

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

3. Group, C.T.: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. NIST CSRC (2016) http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf.
4. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. EUROCRYPT 1999. LNCS **1592** (1999) 206–222
5. Patarin, J., Goubin, L., Courtois, N.: C $^{*}_{-+}$ and HM: Variations around two schemes of T.Matsumoto and H.Imai. Asiacrypt 1998, Springer **1514** (1998) 35–49
6. Patarin, J., Courtois, N., Goubin, L.: Quartz, 128-bit long digital signatures. In Naccache, D., ed.: CT-RSA. Volume 2020 of Lecture Notes in Computer Science., Springer (2001) 282–297
7. Petzoldt, A., Bulygin, S., Buchmann, J.: Cyclicrainbow - a multivariate signature scheme with a partially cyclic public key. In Gong, G., Gupta, K.C., eds.: INDOCRYPT. Volume 6498 of Lecture Notes in Computer Science., Springer (2010) 33–48
8. Petzoldt, A., Chen, M., Yang, B., Tao, C., Ding, J.: Design principles for hfev- based multivariate signature schemes. In Iwata, T., Cheon, J.H., eds.: Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I. Volume 9452 of Lecture Notes in Computer Science., Springer (2015) 311–334
9. Ding, J., Yang, B.Y.: Degree of regularity for hfev and hfev-. [23] 52–66
10. Goubin, L., Courtois, N.: Cryptanalysis of the ttm cryptosystem. In Okamoto, T., ed.: ASIACRYPT. Volume 1976 of Lecture Notes in Computer Science., Springer (2000) 44–57
11. Tsujii, S., Gotaishi, M., Tadaki, K., Fujita, R.: Proposal of a signature scheme based on sts trapdoor. In Sendrier, N., ed.: PQCrypto. Volume 6061 of Lecture Notes in Computer Science., Springer (2010) 201–217
12. Porras, J., Baena, J., Ding, J.: Zhfe, a new multivariate public key encryption scheme. [22] 229–245
13. Tao, C., Diene, A., Tang, S., Ding, J.: Simple matrix scheme for encryption. [23] 231–242
14. Ding, J., Petzoldt, A., Wang, L.: The cubic simple matrix encryption scheme. [22] 76–87
15. Szepieniec, A., Ding, J., Preneel, B.: Extension field cancellation: A new central trapdoor for multivariate quadratic systems. [24] 182–196
16. Perlner, R.A., Smith-Tone, D.: Security analysis and key modification for ZHFE. [24] 197–212
17. Moody, D., Perlner, R.A., Smith-Tone, D.: An asymptotically optimal structural attack on the ABC multivariate encryption scheme. [22] 180–196
18. Moody, D., Perlner, R.A., Smith-Tone, D.: Key recovery attack on the cubic abc simple matrix multivariate encryption scheme. In: Selected Areas in Cryptography – SAC 2016: 23rd International Conference, Revised Selected Papers, LNCS, Springer (2017)
19. Barker, E., Roginsky, A.: Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. NIST Special Publication (2015) http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf.
20. Diene, A., Tao, C., Ding, J.: Simple matrix scheme for encryption (abc). Presentation: PQCRYPTO 2013 (2013) http://pqcrypto2013.xlim.fr/slides/05-06-2013/Diene.pdf.
21. Developers, T.S.: SageMath, the Sage Mathematics Software System (Version x.y.z). (YYYY) `http://www.sagemath.org`.
22. Mosca, M., ed.: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Volume 8772 of Lecture Notes in Computer Science., Springer (2014)
23. Gaborit, P., ed.: Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings. In Gaborit, P., ed.: PQCrypto. Volume 7932 of Lecture Notes in Computer Science., Springer (2013)
24. Takagi, T., ed.: Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings. Volume 9606 of Lecture Notes in Computer Science., Springer (2016)

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Moody, Dustin; Perlner, Ray; Smith-Tone, Daniel.
"Improved Attacks for Characteristic-2 Parameters of the Cubic ABC Simple Matrix Encryption Scheme."
Paper presented at PQCrypto 2017: The Eighth International Conference on Post-Quantum Cryptography,
Utrecht, Netherlands. June 26, 2017 - June 28, 2017.

Analysis, Comparison, and Assessment of Latent Fingerprint Image Preprocessing

Haiying Guan*, Paul Lee*, Andrew Dienstfrey*, Mary Theofanos*, Curtis Lamp*, Brian Stanton*, and Matthew T. Schwarz+
*National Institute of Standards and Technology Gaithersburg, MD, USA
*Email: {haiying.guan, paul.lee, andrew.dienstfrey, mary.theofanos, curtis.lamp, brian.stanton}@nist.gov
+ Schwarz Forensic Enterprises, Inc., +Email: matt@schwarzforensic.com

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Abstract

Latent fingerprints obtained from crime scenes are rarely immediately suitable for identification purposes. Instead, most latent fingerprint images must be preprocessed to enhance the fingerprint information held within the digital image, while suppressing interference arising from noise and otherwise unwanted image features. In the following we present results of our ongoing research to assess this critical step in the forensic workflow. Previously we discussed the creation of a new database of latent fingerprint images to support such research. The new contributions of this paper are twofold. First, we implement a study in which a group of trained Latent Print Examiners provide Extended Feature Set markups of all images. We discuss the experimental design of this study, and its execution. Next, we propose metrics for measuring the increase of fingerprint information provided by latent fingerprint image preprocessing, and we present preliminary analysis of these metrics when applied to the images in our database[1]. We consider formally defined quality scales (Good, Bad, Ugly), and minutiae identifications of latent fingerprint images before and after preprocessing. All analyses show that latent fingerprint image preprocessing results in a statistically significant increase in fingerprint information and quality.

I. INTRODUCTION

Latent fingerprints are friction ridge impressions left unintentionally on the surface of an object. Images of latent fingerprints can be obtained (i.e., "lifted" or "developed") through numerous methods ranging from precision photography to complex physical and chemical processing techniques [1]. Latent fingerprint evidence plays an important role in forensic science and is routinely used as evidence to convict offenders of crimes. From the unintentional deposition and complexity of acquisition, it follows that the initial latent fingerprint images collected directly from a crime scene may be incomplete or hard to visualize, leading to images of very poor quality. Lighting, pressure, and underlying surface qualities such as texture and color are just a few factors that may affect the quality of a fingerprint digital image [2].

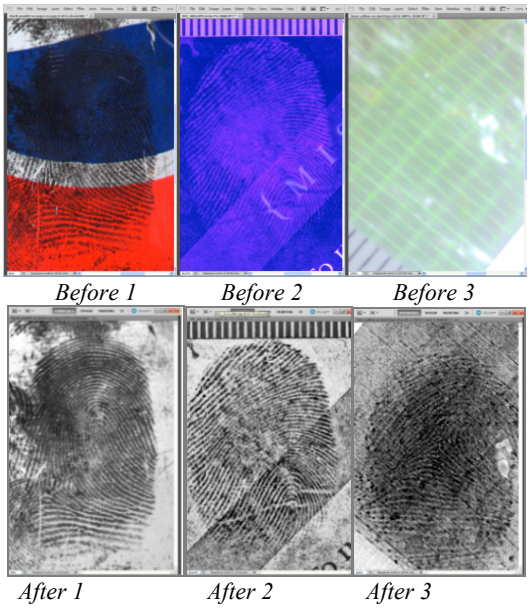


Figure 1: Latent Fingerprint Before and After Preprocessing Examples.

Consider the top row of latent fingerprint examples shown in Figure 1. Due to the low signal quality of the fingerprint in relation to other systematic image features, such as color, pattern, text, etc., the initial fingerprint image quality may be of only marginal value for identification. In some extreme cases, latent prints are identified as "no value." In this context, "no value" is a formal determination that the print is of such poor quality that no identification—neither individualization nor exclusion—is possible. This is true regardless of the score of a potential match between the latent to other prints held in a database [3]. Thus, potentially usable latent images are classified as unsuitable for feature markup, entry into databases, or input into fingerprint identification software to search for matches.

To mitigate this issue, current practice allows for a Latent Print Examiner (LPE) to perform image preprocessing prior to markup and feature analysis. The forensics community currently uses a variety of image analysis and preprocessing tools to significantly improve the quality of these images and enhance fingerprint features. The bottom row of latent fingerprint examples in Figure 1 are the result of preprocessing the images in the top row, which show that the changes can be extraordinary. For example, the ridge patterns are significantly more visible in the first and second images. In the third, a grid-like background has been removed to

[1] The latent fingerprint images are from a training data set provided by the course from FORAY technologies and Schwarz Forensic Enterprises, Inc.

Dienstfrey, Andrew; Guan, Haiying; Lamp, Curtis; Lee, Paul; Schwarz, Matthew; Stanton, Brian; Theofanos, Mary.
"Analysis, Comparison, and Assessment of Latent Fingerprint Preprocessing."
Paper presented at CVPR 2017, Honolulu, HI. July 22, 2017 - July 25, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

reveal fingerprint information "underneath". In short, latent fingerprint image preprocessing can transform raw images with little or no value into ones suitable for evidentiary analysis.

Some of the key components of evidentiary analysis such as automatic fingerprint feature extraction, matching, and print type identification are well studied, regulated, and implemented in existing systems. However, the preprocessing step is currently overlooked. Preprocessing is the first step of the analysis workflow, and can be critical to the accuracy of subsequent analysis [4]. For instance, an image with extraneous noise introduced during the preprocessing phase may lead to incorrect feature extraction, which may have a negative effect during the matching and identification stages. Despite the importance of this step, there exist few databases for controlled experimentation and scientific study, and even fewer standards. Detrimental consequences for reproducibility, traceability, and quantification of accuracy naturally follow. Our research hopes to shed some light on this topic.

Previously Guan et al. [4] presented the results of the collaboration with forensic scientists to design and collect a database of latent fingerprint images consisting of: original latent fingerprint images ("Before"), their pre-processed counterparts ("After"), and documentation of the image transformation procedures executed during the preprocessing stage. The paper also proposed a new latent print quality measurement metric. Here we extend this previous work in two significant ways. First, we designed a round-robin experiment contracting an independent set of LPEs certified by the International Association for Identification to provide Extended Feature Set (EFS) [5] markups for all images. Such markup information significantly increases the value of this database. Next, we conducted experiments analyzing the fingerprint data quality in the several image classes within the database. More specifically, we use three metrics—value determination, minutiae count, and quality confidence score—to compare changes in image quality and fingerprint information that result from preprocessing. We find that examiners mark more minutiae on the After latent image than the Before. Additionally, examiners categorize the preprocessed images higher on a quality scale, resulting in an improved quality confidence scores as compared to the Before images. Finally, our analysis shows that LPEs identify more minutiae in color images than in grayscale (i.e., "Before Color" as compared to "Before Gray"). This suggests that there may be value in having color images available for input in Automated Fingerprint Identification Systems (AFIS).

In summary, we intend that these results will provide foundational elements for a systematic and scientific basis for latent fingerprint analysis. Furthermore, we hope that they may serve as a test case for the development of comparable analysis for other image-based methods in forensic science in the future.

II. METHODOLOGY

2.1 Study Objective

The objective of this study is to determine the quantitative value of latent fingerprint images before and after preprocessing, focusing on notable changes in detectable fingerprint minutiae. Additionally, we are also looking at quality scale changes and quality map changes in color vs. grayscale images. Note that unlike other studies [3][6], this study does not compare the markups among examiners nor evaluate examiners' performance. No identifying information is kept or linked to the images. The chief goal is to determine whether and to what extent latent fingerprint preprocessing improves the ability to gain information in the identification of latent impressions, as well as to what extent it transforms latent images with no comparison value into images that can be used for analysis.

2.2 Initial Dataset

Previously we created a database of latent fingerprint images isolating several steps within the preprocessing workflow. This database includes 89 latent fingerprint image pairs that were developed using a cross-section of forensic field work techniques including: ninhydrin, silver magnesium powder, white powder, bi-chromatic powder, bi-chromatic mag powder, and black ink. The original images were scanned by high-resolution flatbed scanners and subsequently preprocessed within Adobe Photoshop, the primary image analysis tool used by latent examiners practicing today.[2] The image transformations in the preprocessing workflow were recorded in Adobe Photoshop and saved in an accompanying metadata file as per existing best-practice guidelines [7]. The result was a collection of triplets consisting of original image, processed image, and metadata file. This database has proven to be an invaluable source of controlled data for developing scientific analyses of forensic image preprocessing.

2.3 Experimental Design

A team of 9 independent LPEs was assembled. Images were distributed and presented to the LPEs in a pre-determined order per the following assignment criteria:

- Examiners receive and mark up one image at a time. No information regarding whether the image has undergone preprocessing is given to the examiner.
- Generally, an examiner will not mark up the After image corresponding to any previously seen Before image.

[2] Any mention of commercial products or reference to commercial organizations is for information only. It does not imply recommendation nor endorsement by NIST, nor does it imply that the products mentioned are the best available for the purpose.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

- Each examiner receives at least one good quality image and one bad quality image. The remainder will be a mix of good, bad, and ugly images.
- Finger source distribution is randomized, ensuring approximately the same distribution amongst examiners. The study has three phases:

Phase I: Examiners mark Before Grayscale images.
Phase II: Examiners mark the corresponding Before Color images. It is acceptable for an examiner to view the Grayscale image to assist in this markup process.
Phase III: Examiners mark After Grayscale images.

In each phase, the examiners were given a list of images to analyze. The results of the previous phase were collected prior to release of the next phase's image set. We implemented a sorting algorithm to assign the images to the examiners in different phases, attempting to satisfy the above design criteria as best as possible.

When performing the markup, LPEs assume the images provided are the only images available, and that physical evidence, lift cards, fingerprint cards, additional exemplars, and different images of these prints are not available. For consistency, LPEs use Universal Latent Workstation Latent Editor software, ULW-EFS 6.4.0 or newer (https://www.fbibiospecs.cjis.gov/Latent/PrintServices), to do markup. Each received a standardized instruction document on how to proceed at all stages of the study to guide their work.

Upon receiving an image, the LPEs were required to perform the following steps: (1) Paint the quality (clarity) of the latent (throughout the entire region of interest), (2) Annotate EFS features within the image, and (3) Record the final value impression determination of each print using the Good, Bad or Ugly (GBU) quality scale [2].

The EFS was developed by Noblis (http://www.noblis.org) in collaboration with the Federal Bureau of Investigations and standardizes the diverse fingerprint image metadata considered useful for identification analysis. The EFS augments the ridge-flow information contained within a fingerprint image by inserting standardized indications of features including: ridge quality maps, incipient ridges, minutiae, cores, deltas, and others. LPEs followed instructions of the ACE-V (Analysis, Comparison, Evaluation, and Verification) methodology to assess images for the presence of: friction ridges, fingerprint information available, the confidence of such information etc. Enhancement tools present in the ULW Latent Editor software or in any other software that the examiner might have available were strictly forbidden. Under our study, three versions of each latent image were marked by examiners and the EFS information are held within the database: the original Before Color image, the original Before Grayscale image, and the preprocessed After image. Each latent image was marked by two different LPEs.

In addition, we collected rolled print images and performed EFS markups for every finger source. The markup of these prints allows us to furnish "ground truth" EFS data, which serves as a basis of comparison between Before and After images. Prior to comparing markups of a latent image to its corresponding rolled image, the two images must be aligned. In forensic practice, such image registration is accomplished as a sub-task of EFS feature comparison. In the present study, we sought to eliminate this source of variability. For each latent record an independent examiner identified a number (>3) of benchmark minutiae that could be found on both the latent and its associated ground-truth. Corresponding minutiae were indicated by color. Ideally these features are as separated as possible throughout the region of interest. A color point detection algorithm identified the locations of corresponding features, and a least-squares algorithm was used to estimate the rigid transformation parameters (rotation and translation) to transform ground-truth orientation to that of the latent.

III. LATENT PREPROCESSING DATABASE

The latent preprocessing database contains 89 fingerprint records. Structurally, each record is a directory containing: several image files, their EFS markups saved in the Latent Friction Features Search format (defined by the Electronic Biometric Transmission Specification described in https://www.fbibiospecs.cjis.gov/ebts/), the source finger's card image and its EFS markup, and metadata files. The various metadata files include: examiner ID, source finger ID, GBU value determinations, image resolutions, specific latent lifting techniques, etc. In total, there are 28 files for each record. We describe these in more detail below.

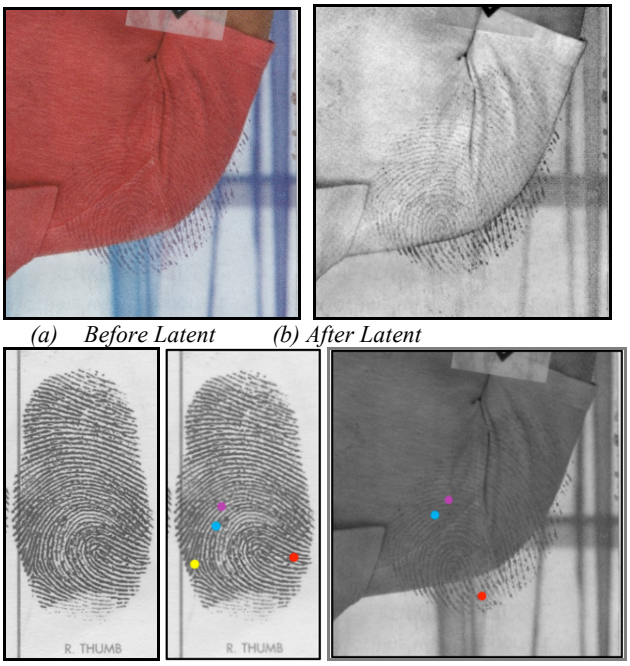
3.1 Image files

The Before latent image is the latent fingerprint scan that has yet to undergo preprocessing. Note that while preprocessing is performed in Adobe photoshop on high-resolution images scanned at 1200ppi, the ULW-EFS 6.4.0 requires images to be 1000ppi. Thus there are three Before image files in our dataset: the original Before latent color image in its native scanned resolution of 1200ppi, its downscaled color version of 1000ppi, and its downscaled grayscale version of 1000ppi. All down sampling was done using OpenCV (http://www.opencv.org/) bicubic interpolation.

Once the original Before Color image scan at 1200ppi undergoes preprocessing, the After Grayscale image is obtained. Unlike the Before category with its grayscale and color versions, After images are only in grayscale. These high contrast versions of latent images are commonly used in AFIS [8] search or matching; the grayscale property is required by this system. In the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

database, there are two After files: one latent image at the native scanned resolution (1200ppi) and one scaled down to 1000ppi. Once again, the 1000ppi image is required by the ULW software. Figure 2 (a) and (b) show a sample pair of a Before Color latent image and its After preprocessed image.



(a) Before Latent *(b) After Latent*

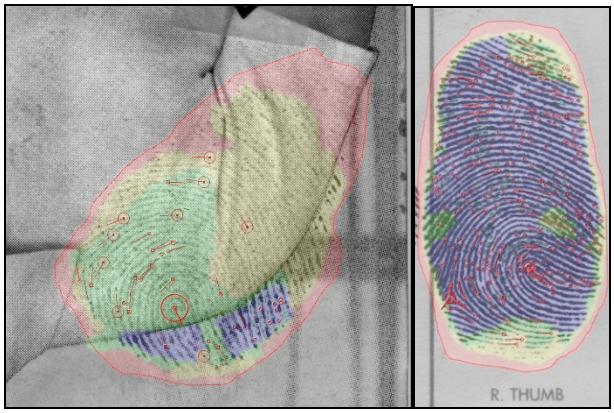
(c) Card (d) Card Minutiae Annotation (e) Latent Minutiae Annotation

Figure 2: Images in latent preprocessing database

We also collect the source finger's card image and its EFS markup file as the reference ground-truth for the latent images. Figure 2 (c) is an example of the original, unmarked finger source's card image. To align the minutiae in latent image with the minutiae in card image, the latent examiner uses colored dots to annotate at least three minutiae in the card image and in the After image (Figure 2 (d) and (e)). Note that the color dot radius is enlarged for illustration purposes. The actual dot radius within the image is 5 pixels at 1200ppi.

3.2 Markup Files

Following the experimental design, in the first round two latent print examiners mark up the EFS Before Color and Before Grayscale images. Examiners inspect and mark miscellaneous minutiae, bifurcations, incipient ridges, ridge endings, dots, the region of interest, and distinctive quality areas. Bifurcations are marked with squares, incipient ridges by green lines, ridge endings by small circles with trailing tails, and deltas by large circles with bisected centers. Minutiae too obscure to classify are represented by lone circles of two sizes to represent the uncertainty, with higher quality unknown minutiae corresponding to the smaller of the two. When the first round is complete, After Grayscale images are released. Two examiners repeat the process for the preprocessed images, marking all the same feature categories. In total, seven EFS markup files in the EBTS format (.lffs) are collected: a markup of the card image, and two independent markup files for the Before Grayscale, Before Color, and After Grayscale latent images. Each of these markup files are accompanied by a corresponding text document containing data such as: minutiae coordinates, minutiae types, the quality map matrix, image metadata, etc. The After Grayscale markup of Figure 2(b) is shown below in Figure 3(a). The markup of the card(source) image is also collected as shown in Figure 3(b).



(a)After Grayscale Markup *(b) Card(Source)Markup*

Figure 3: Full Markup Images

Latent Quality Mapping is used to document the level of confidence in the marked features. Image quality is documented by painting over the image using standard color definitions for latent region quality markup. The color scale range includes cyan, blue, green, yellow, red, and black, in order of the largest level of confidence to the smallest. Teal indicates that there are clear definitive ridge edges plus, dots, pores and level three detail throughout the area, blue indicates that there are clear ridges, and green indicates that it is certain that every minutia in the area is marked. Note that green (or better) means that the examiner is certain of the presence of all minutiae they've marked in that region AND they are certain that there are no unmarked minutiae. Yellow indicates that the examiner is not confident in the presence or location of marked minutiae and there may be minutiae in the area that they did not mark. Finally, red indicates any discontinuities (e.g., smears), and black indicates the lack of ridge data in a particular area of the image. The Latent Quality Markup of Figure 2 can be seen above in Figure 3. Additional information about the ridge quality map and feature markup on this study can be found in the ANSI/NIST standard [9] and Markup Instructions for Extended Friction Ridge Features [7].

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3.3 Metadata

Alongside the various images, six metadata files are also included. The first information spreadsheet contains most experimental design related details, including the source fingerprint, the examiner IDs, the latent acquisition procedure used, the GBU classification of the Before and After files, and the various image resolutions. The second information spreadsheet holds relational database details. The action history of the preprocessing editing session are recorded in a word document, including file creation, color channel selection, color scheme conversion, use of the burn tool, etc. The last three files are all single item files that hold the translational matrix for the latent image to card image shift, the manually annotated rectangular region of interest coordinates, and the manually annotated polygon coordinates respectively.

IV. ANALYSIS

Data points represented here include the 89 sample records. The significance of preprocessing was determined by analysis of changes in three quality metrics: image value determination, minutiae count, and quality confidence score.

4.1 GBU Value Determination Comparison

Upon receiving the latent image, examiners determine the overall latent quality using the Good, Bad, and Ugly scale [2]. After pre-processing, the images were examined again and re-categorized. Table 1 shows the number of images classified as Good, Bad and Ugly for the Before and After datasets as well as the change in quality scale determination after preprocessing.

Table 1: Value Determination and Re-Categorization

Before \ After	Good	Bad	Ugly	Total
Good	25	0	0	25
Bad	23	12	0	35
Ugly	5	16	8	29
Total	53	28	8	89

Across each Before row (Before Good, Before Bad, and Before Ugly), the number of images initially classified as such are noted. Of the 89 records, 25 of the Before images were classified as Good, 35 were classified as Bad, and 29 were classified as Ugly. Down each After column (After Good, After Bad, and After Ugly), the number of images re-classified into these categories can be seen. Of the 64 images previously categorized as either Bad or Ugly, 28 were assessed to be of Good quality after preprocessing (Bad: 23, Ugly: 5). The remaining 36 images were split between Ugly images rising to Bad quality (16), and images in both categories staying in their initial determination category (Bad: 12, Ugly: 8). Each of the 25 initial Good quality latent images remained in the Good category. No instances of quality deterioration were found.

These tables show that across quality determination categories, 49.43% of latent fingerprint images showed marked improvement after preprocessing. Excluding the images initially rated Good (as these cannot be improved), we find that 68.75% are improved by preprocessing.

4.2 Minutiae analysis

In our collection, we have three types of latent images: Before Color, Before Grayscale, and After Grayscale. Each latent image was reviewed and marked by two examiners. The original .lffs files were fed through our minutiae reader tool to analyze minutiae markup data. After accounting for differences in resolution, horizontal and vertical offsets, and verifying miscellaneous examiner markups, the result were 6 sets of minutiae coordinates aligned in the same coordinate system: two for Before Color, two for Before Grayscale, and another two for After Grayscale.

To compensate for possible differences between examiners, we designed and implemented an algorithm to identify corresponding minutiae between two markups of the same image. Minutiae correspondence was determined by procedure involving a combination of: minutiae proximity, distance hierarchy assignments, minutiae type, and final manual verification examinations to guarantee the correctness. For a pair of markup files of the same image, this intersection set represents a consensus understanding of an image's feature set, with a singular representation of each minutiae. Unmatched minutiae are retained in the database but are not included as part of the following analysis.

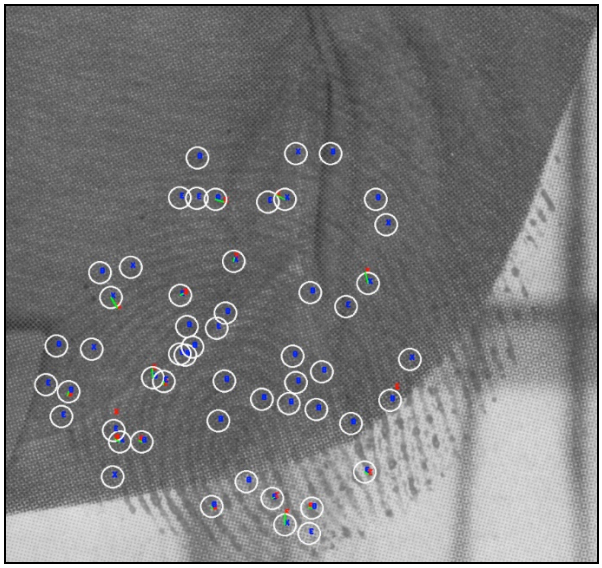


Figure 4: Before Gray and After Grayscale Intersection Minutiae

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

In Figure 4, both intersection minutiae sets of Before Gray and After Grayscale of Figure 2 can be seen. The Before Gray intersection set of Figure 2 is shown in red, while the After-Grayscale intersection set is shown in blue. The white circle around each After Grayscale minutiae represents the "search area" used to seek out matching Before minutiae, with green lines marking a successful match. Different types of minutiae are marked with different letters, with 'E' used for endpoints, 'B' used for bifurcations, and 'X' used for unclassified minutiae.

We present the results based off the intersection data set below.

4.2.1 Percentage Gain

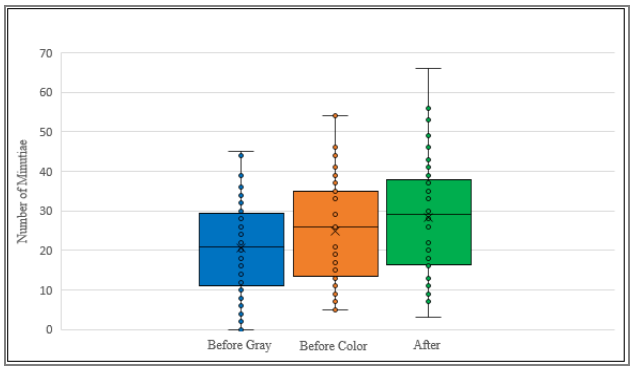


Figure 6: Minutiae Count

Overall minutiae feature count across image types is shown in Figure 6. There was an average of 20.74 minutiae in the Before Grayscale images, 24.82 minutiae in the Before Color images, and 28.44 minutiae in the After images.

The corresponding median minutiae gain percentages are shown in Table 2. With the most commonly used AFIS systems and matching tools limited to or being heavily reliant on grayscale images, only the After latent image in grayscale is currently available to be studied. In the future, based on our experiments, we suggest that After (preprocessed) images in color may also help preserve useful feature information.

Table 2: Minutiae Gain Percentage

Image Comparison	Median	Mean
BG to BC	14.84%	30.55%
BG to AG	34.59%	70.39%
BC to AG	8.82%	30.92%

For each latent image pair, we calculated the increase in minutiae count from the Before image to the After image, and divided by the Before image minutiae count. We then derived the mean and median gain percentages across the entire dataset using this series of percentages. Comparing Before Grayscale to Before Color latent images results in a 14.84% increase in minutiae found. Before Grayscale to After measures in at a 34.59% increase, while Before Color to After results 8.82% increase. Mean percentage gain comes in much higher across the board, with increases of 30.55% for Before Grayscale to Before Color, 70.39% for Before Grayscale to After, and 30.92% for Before Color to After.

The larger outliers in the gain percentage distribution contribute to the differences between median and mean gain, as seen in Figure 7. The figure on the right presents the same graph, but with a reduced scale for easier viewing. The figure only covers two out of the four comparison categories, but the distribution is similar for all four: majority clusters from about -40% to 100%, then decreasing distribution until about 200%, with a few outliers of more than five times the original minutiae count.

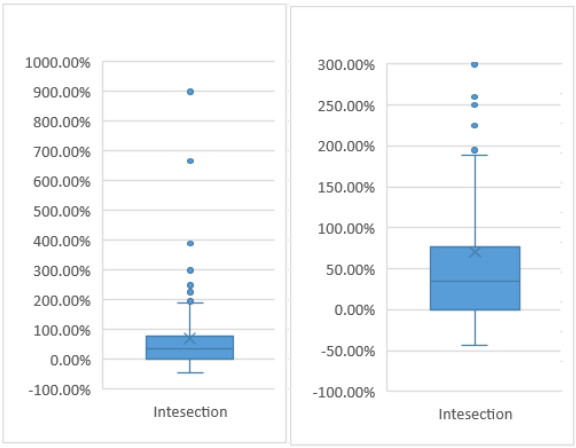


Figure 7: Minutiae Gain Percentage Distribution

4.2.2 Signed Ranks Test

The Wilcoxon Signed Rank Test [10] was used to test the significance of differences in minutiae counts between treatment groups, for example, Before Grayscale to After. Given a paired list of minutiae counts, this nonparametric test computes a score by: 1. rank ordering the absolute value of all differences, 2. reassigning the sign of the difference to the ranked list, and 3. evaluating the signed rank sum (W). Under the null hypothesis that the minutiae count distribution is the same between the two groups, W will be close to zero. Considering this comparison with zero, as all sample sizes are greater than 10, W may be approximated by a normal random variable. We calculate the z-value by dividing the critical value (W) by the standard deviation of its sampling distribution (σw). The standard deviation is derived by taking the square root of $(N_r(N_r+1)(2N_r+1))/6$, where N_r is the sample size.

$$z = \frac{W}{\sigma w}, \sigma w = \sqrt{\frac{N_r(N_r+1)(2N_r+1)}{6}} \quad (1)$$

Based on the z-value, we can determine the two-tailed probability score P (<0.05). This description is brief. For

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

more details see [10].

Like the minutiae gain analysis, all significance tests compared the relationship between Before Grayscale and Before Color latent images, Before Grayscale latent image and After images, and Before Color and After images, using the 89 previously used records. Note the different values of N_r are the result of image pairs with no change in the number of minutiae features identified. In these cases, the sample is not included in the significance analysis. The results can be seen below in Table 3.

The first comparison was done to measure any marked improvement between image qualities of the starting sample. The Before Grayscale to Before Color comparison results in W=-2507 and a Z value of -6.13, leading to a P=<.0001. The second comparison was performed to measure any improvement in minutiae detection due to pre-processing. Comparing Before Grayscale to After, there are 87 samples with W=-2769, resulting in a Z value of -5.86 and P=<.001. The final comparison measures the ability of the Before Color image to preserve feature information. With 85 samples, Before Color to After has W=-1451, Z=-3.18, and P=.0015.

Table 3: Wilcoxon Signed Ranks Test - Minutiae Count

Comparison	N	Test Statistic(W)	Z	P
BG to BC	79	-2507	-6.13	<.0001
BG to AG	87	-2769	-5.86	<.0001
BC to AG	85	-1451	-3.18	0.0015

With α=.05, the Wilcoxon Signed Rank Test indicates that differences between all treatments groups in Table 3 are statistically significant. In other words, more minutiae are identified in the Before Color than their grayscale counterparts (BG to BC), and also, the preprocessed images contain more minutiae than either of the before images (BG to AG, and BC to AG). We note that we also tested for the significance of minutiae count differences using a Random Matched Sample analysis which confirmed the results shown here.

4.3 Quality Confidence Score

In addition to the improvement in image quality that can be derived from GBU value determination and minutiae count gain, we also assess image quality gains by looking at the differences in the quality confidence score of image treatments. The quality confidence score of each markup image is derived by cross referencing the coordinate position of each marked minutiae with the Latent Quality Mapping of the image as detailed in section 3.2. The quality confidence score of a given latent fingerprint is defined as follows: given each marked minutia in the latent image, locate its position in the quality map, and obtain its quality map value given that position. Then we sum up all minutia quality map values and obtain a final single quality confidence score for the image. The quality confidence score measures how thoroughly a LPE could mark the features of the latent image, as well as how confident they are in specific minutiae locations. More minutiae or larger areas of higher quality results in a higher overall quality score.

Each record's quality confidence score gain percentage was collected and averaged, resulting in a mean score gain of 22.81% when comparing Before Grayscale to Before Color, and 23.55% when comparing Before Grayscale to After. Median score gains were slightly more detached, with a 20.00% gain for Before Grayscale to Before Color and 29.38% for Before Grayscale to After. To measure the significance of the result, we again used the Wilcoxon Signed Ranks Test.

Of the 89 records previously used in quality comparisons, 86 were used in conjunction with the Wilcoxon Signed Ranks Test to determine quality change significance between Before Grayscale and Before Color images. Of the three unused records, two were removed due to missing minutiae markup, while the third had the same quality confidence score for both the Before Grayscale and Before Color images. For the Before Grayscale to After comparison, 87 out of the 89 records were used. The two unused records were the same two removed in the Before Grayscale to Before Color comparison due to missing minutiae markup. This holds true for the Before Color and After comparison as well. The results of the test can be seen in Table 5.

Table 5: Wilcoxon Signed Ranks Test - Quality Confidence Score

Comparison	N	Test Statistic(W)	Z	P
BG to BC	86	-3456	-7.440	<.0001
BG to AG	87	-2886	-6.107	<.0001
BC to AG	87	-1014	-2.05	.0324

The first comparison, Before Grayscale to Before Color, has the test statistic (W) = -3456 and the Z value = -7.440, which leads to P = <.0001. The second comparison, Before Grayscale to After, has the test statistic (W) = -2886 and the Z value = -6.107, which leads to P = <.0001. The third, Before Color to After, has the test statistic (W) = -1014 and the Z value = -2.05, which leads to P = .0324. With α=.05, the Wilcoxon Signed Ranks test indicates that the differences in quality score between the Before Grayscale and Before Color, the Before Grayscale and After, and Before Color to After are all statistically significant.

V. CONCLUSION

Currently many prints that could be preprocessed through software are not analyzed or compared because they are deemed "no value" [11]. Furthermore, due to the lack of quantitative techniques for image preprocessing many forensic laboratories currently do not employ or allow

image preprocessing software. We hope that the dataset discussed in this paper, complete with images and EFS markups, will provide forensic analysts a testbed for future preprocessing studies.

Along these lines, we designed a series of comparison experiments to examine the effectiveness of preprocessing in relation to feature marks. The quantitative results show that the After latent image is significantly improved by preprocessing. While the scope of this paper is limited to latent fingerprints preprocessing, the design approach and analyses methods are applicable to other biometrics comparative disciplines including handwriting, footwear, tool marks, tread marks, firearm compressions, bite marks, bruising, and so on.

Future work involving the database will include another series of comparisons using the quality map feature. We will also continue to provide techniques and processes enabling latent fingerprint examiners to analyze and compare evidence more effectively, as well as build foundations for future academic research and standards formulation.

REFERENCES

[1] H. C. Lee and R. E. Gaensslen, Eds., *Advances in fingerprint technology*, 2nd ed. Boca Raton, Fla: CRC Press, 2001.

[2] R. Rajkumar and K. Hemachandran, "Latent Fingerprint Enhancement in Preprocessing Stage," in *International Conference on Computer Science and Intelligent Systems (CSIS'11)*, 2011.

[3] B. T. Ulery, R. A. Hicklin, G. I. Kiebuzinski, M. A. Roberts, and J. Buscaglia, "Understanding the sufficiency of information for latent fingerprint value determinations," *Forensic Sci. Int.*, vol. 230, no. 1–3, pp. 99–106, Jul. 2013.

[4] H. Guan, A. M. Dienstfrey, and M. F. Theofanos, "A New Metric for Latent Fingerprint Image Preprocessing," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 84–91.

[5] M. Taylor *et al.*, "Extended Feature Set Profile Specification," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 1134, Dec. 2012.

[6] B. T. Ulery, R. A. Hicklin, M. A. Roberts, and J. Buscaglia, "Measuring What Latent Fingerprint Examiners Consider Sufficient Information for Individualization Determinations," *PLOS ONE*, vol. 9, no. 11, p. e110179, Nov. 2014.

[7] P. Peterson *et al.*, "Latent Prints: A Perspective on the State of the Science," *FBI Forensic Sci. Commun.*, vol. 11, no. 4.

[8] L. M. I. Dexter and A. J. Pouratian, "Automated fingerprint identification system," US5869822 A, 09-Feb-1999.

[9] K. C. Mangold, "Data Format for the Interchange of Fingerprint, Facial & Other Biometric Information ANSI/NIST-ITL 1-2011." NIST Special Publication 500-290 Edition 3, 22-Aug-2016.

[10] R. Lowry, "Concepts and Applications of Inferential Statistics, by Richard Lowry," *Online Book*, 2015. [Online]. Available: http://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp66608. [Accessed: 03-Mar-2017].

[11] M. Theofanos, B. Stanton, D. Witzke, and M. Schwarz, "Characterizing the Latent Fingerprint Pre-Processing Procedures," National Institute of Standards and Technology, NIST Internal or Interagency Reports (NISTIR), 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Securing Networks against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options

Daniel Borbor[1], Lingyu Wang[1], Sushil Jajodia[2], and Anoop Singhal[3]

[1] Concordia Institute for Information Systems Engineering, Concordia University
{d_borbor,wang}@ciise.concordia.ca
[2] Center for Secure Information Systems, George Mason University
jajodia@gmu.edu
[3] Computer Security Division, National Institute of Standards and Technology
anoop.singhal@nist.gov

Abstract. The administrators of a mission critical network usually have to worry about non-traditional threats, e.g., how to live with known, but unpatchable vulnerabilities, and how to improve the network's resilience against potentially unknown vulnerabilities. To this end, network hardening is a well known preventive security solution that aims to improve network security by taking proactive actions, namely, hardening options. However, most existing network hardening approaches rely on a single hardening option, such as disabling unnecessary services, which becomes less effective when it comes to dealing with unknown and unpatchable vulnerabilities. There lacks a heterogeneous approach that can combine different hardening options in an optimal way to deal with both unknown and unpatchable vulnerabilities. In this paper, we propose such an approach by unifying multiple hardening options, such as firewall rule modification, disabling services, service diversification, and access control, under the same model. We then apply security metrics designed for evaluating network resilience against unknown and unpatchable vulnerabilities, and consequently derive optimal hardening solutions that maximize security under given cost constraints.

1 Introduction

Today's computing networks are playing the role of nerve systems in many mission critical infrastructures, such as cloud data centers and smart grids. However, the scale and severity of security breaches in such networks have continued to grow at an ever-increasing pace, which is evidenced by many high profile security incidents, such as the recent large scale DDoS attacks caused by the Mirai Botnet on the Dyn DNS, and the cyber-physical attack on Ukraine power grid in 2015. The so-called zero day attacks, which exploit either previously unknown or known, but unpatched vulnerabilities, are usually behind such security incidents, e.g., Stuxnet employs four different zero day vulnerabilities to target SCADA. Therefore, administrators of a mission critical network usually need to worry about not only patching known vulnerabilities and deploying traditional defense mechanisms (e.g., firewalls, IDSs, and IPSs), but also non-traditional security threats, e.g., how to live with known, but unpatchable vulnerabilities, and how to improve the network's resilience against potentially unknown vulnerabilities.

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

In fact, it is well known that both cybercriminals and governmental agencies stockpile vulnerabilities that are not publicly known (e.g., the NSA reportedly spent more than 25 million a year to acquire software vulnerabilities, and private vendors are providing at least 85 zero-day exploits on any given day [17]). On the other hand, even for known vulnerabilities, patching is not always a viable option. For example, a patch may not be readily available at the time of the attack, or the system may have reached their end-of-support with no more patch available; patching a vulnerability may cause unacceptable service disruptions on a regular basis (e.g., Windows updates); even worse, patching a vulnerability may sometimes reintroduce other security vulnerabilities that have previously been fixed (e.g., Apache MINA SSHD 2.0.14 introduces an SSL regression previously fixed in 2.0.13 [21]).

Consequently, security professionals need to block the exploitation of such vulnerabilities through other means, such as modifying firewall rules, service diversification, or access control. A critical question is *How to optimally combine such options in order to both improve the security and lower the cost?* To this end, network hardening is a well known preventive security solution that aims to improve network security by taking proactive actions, namely, hardening options. However, most existing network hardening approaches rely on a single hardening option, such as disabling unnecessary services [9, 22] or service diversification [6] (a detailed review of related work will be given later in Section 5). Such a solution becomes less effective when it comes to dealing with unknown and unpatchable vulnerabilities. There lacks a heterogeneous approach that can combine different hardening options in an optimal way to deal with such vulnerabilities.

Running Example We first consider a concrete example to demonstrate why deriving an optimal hardening solution with heterogeneous hardening options would demand a systematic and automated approach. Figure 1 shows a hypothetical network roughly based on Cisco's cloud data center concept [5] as well as the OpenStack architecture [12]. Despite its relatively small scale, it mimics a typical cloud network, e.g., the client layer connects the cloud network to the internet through the CRS 7600; a firewall (ASA v1000) separates the outside network from the inner one. There is a security/authentication layer (authentication server, Neutron server, etc.) as well as a VM and Application layer (Web and application servers). Finally, a storage layer is separated and protected by another firewall (ASA 5500) and an MDS 9000.

We make following assumptions about the network. We assume the two firewalls and other host-based security mechanisms (e.g., personal firewalls or iptables) together enforce the connectivity described inside the connectivity table shown in the figure. External users (including attackers) are represented with host $h0$, and the most critical asset is assumed to be the Xen database server ($h4$), which may be accessed through the three-tier architecture involving hosts $h1$, $h2$, and $h3$. We assume the network is free of any known vulnerabilities, except for an unpatchable vulnerability on the application server running SecurityCenter 5.5 (which cannot be changed due to functionality requirements), and another one on the database server running MySQL 5.7 which may be changed to MSQL 2012 or PostgreSQL 9. For simplicity, we exclude exploits and conditions that involve firewalls in this example.

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

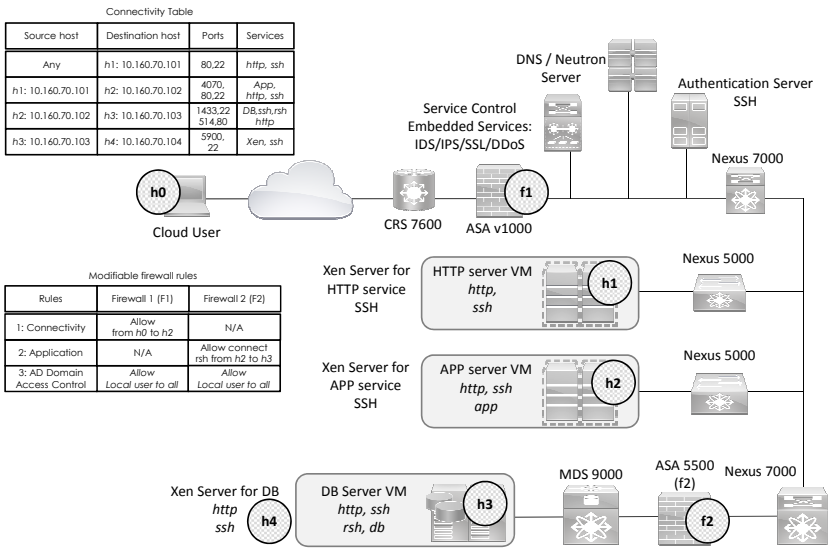


Fig. 1. An Example Cloud Network.

To measure the network's resilience against zero-day attacks, we apply the *k-zero-day safety metric* ($k0d$) [26]. This metric basically counts how many distinct services must be compromised using unknown vulnerabilites before an attacker may compromise the critical asset (i.e., the number of distinct services along the shortest path). In addition, we refine the metric by taking into consideration the potentially uneven distribution of distinct services along the shortest path [30, 33] (e.g., a path consisting of three $http$ and one Xen would be considered slightly "shorter", or less secure, than a path consisting of two $http$ and two Xen, although both paths have the same number of resource instances and resource types).

For hardening options, we consider changes of both the firewall rules and service types. First, we assume the administrator may enable or disable firewall rules on both the firewall ASA v1000 ($f1$) and on the firewall ASA 5500 ($f2$). On $f1$ he has a rule that allows the connection from the cloud user ($h0$) to the app VM ($h2$); he also has the option to allow local user access to $h1$ and $h2$. The firewall ASA 5500 ($f2$) has a rule where he allows the rsh connection on $h3$ from $h2$, as well as local user access to $h3$ and $h4$. Second, we assume the administrator has the option of replacing the Apache Mina 2.0.14 ssh servers with either Copssh 5.8, OpenSSH 7.4, or Attachmate 8.0; the Web servers with either Apache 2.4, IIS 8.5, NGINX 1.9 or a Litespeed 5.0.14 Web server; the rsh service only uses MVRSHD 2.2.

Clearly, even with such a small scale network, the administrator now faces a number of hardening options, including disabling service instances, diversifying service types, and changing firewall rules, each of which may incur certain installation/maintenance cost (we will discuss the cost model in more details later in Section 2). To maximize the resilience of the network against both unknown and unpatchable vulnerabilities, the administrator must decide what would be the optimal combination of such harden-

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

ing options in order to maximize the aforementioned security metric, while respecting given cost constraints. Such a task would obviously be tedious and error prone, if done manually, and demands a systematic and automated approach.

In this paper, we develop such an approach to optimally combine heterogeneous hardening options in order to increase a network's resilience again both unknown and unpatchable vulnerabilities under various cost constraints. Specifically, we first devise our model of different hardening options, costs, and the security metric. We then develop optimization and heuristic algorithms to derive optimal hardening solutions under given cost constraints. We evaluate our approach through simulations in order to study the effect of optimization parameters on accuracy and running time, and the effectiveness of optimization for different types of networks. In summary, the main contribution of this paper is the following.

- To the best of our knowledge, this is the first effort on network hardening using heterogeneous hardening options.
- In constrast to previous works, we provide a refined security metric and an improved cost model that takes into account real world variables in calculating hardening costs.
- Our method is practically relevant to the defense of mission critical networks in which unknown and unpatchable vulnerabilities are realistic security concerns.

The remainder of this paper is organized as follows: In Section 2, we present the model and formulate the optimization problem, and in Section 3 we discuss the methodology and show case studies. Section 4 shows simulation results. Section 5 reviews related work and Section 6 concludes the paper.

2 The Model

We first introduce the extended resource graph model to capture network services and their relationships, then we present the heterogeneous hardening control and cost model, followed by problem formulation.

2.1 Extended Resource Graph

To model network services and their relationships, we revise the *Extended Resource Graph* concept introduced in our previous work [6] in order to model both unpatchable and unknown vulnerabilities, as well as heterogeneous hardening options. The extended resource graph of the running example is shown in Figure 2 and detailed below.

In Figure 2, each pair shown in a rectangle is a security-related condition. If the condition is a privilege, it is represented as $\langle privilege, host \rangle$; if it is connectivity, it is represented as $\langle source, destination \rangle$. If a firewall affects a security-related condition, it is represented as $\langle privilege, firewall, host \rangle$ or as $\langle source, firewall, destination \rangle$. Each one of the rows below the rectangle indicate different hardening options available for that condition. The option currently in use is indicated by the highlighted integer (e.g., 0 means disabled; in the case of service diversification, 1 means Apache, and 2

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

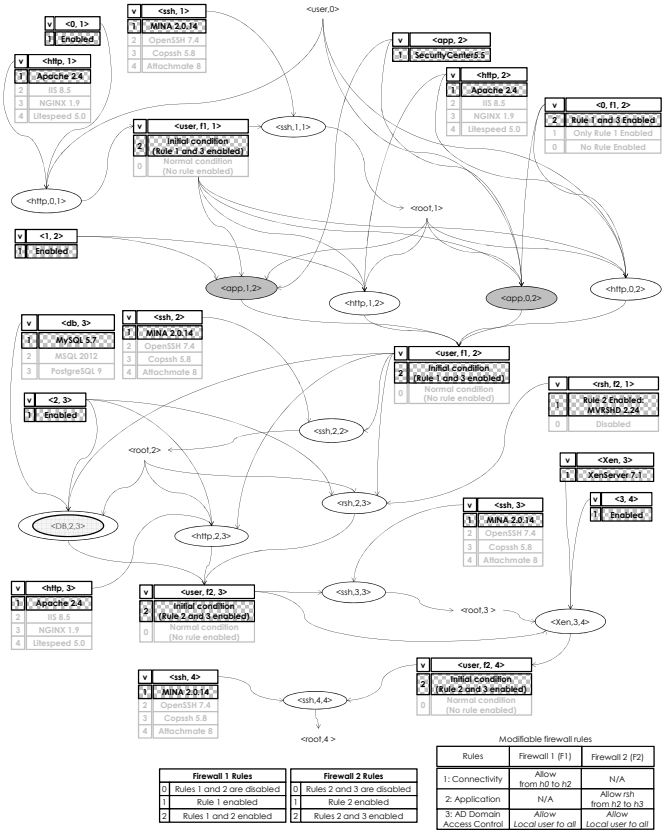


Fig. 2. The extended resource graph of our running example.

means IIS) and other potential instances are in a lighter text. For the conditions modifiable by a firewall rule, the rows below the rectangle indicate the firewall rules that affect it.

Each exploit node (oval) is a tuple that consists of a service running on a destination host, the source host, and the destination host (e.g., the tuple $\langle http, 1, 2\rangle$ indicates a potential zero-day vulnerability in the $http$ service on host 2, which is exploitable from host 1). If the exploit is unpatchable, but diversifiable, it is represented by a double oval; if it is neither patchable nor diversifiable, it is represented as a colored oval (those different types of exploits will contribute to the calculation of the security metric value, as detailed later). The self-explanatory edges point from preconditions to an exploit (e.g., from $\langle 0, 1\rangle$ and $\langle http, 1\rangle$ to $\langle http, 0, 1\rangle$), and from the exploit to its post-conditions (e.g., from $\langle http, 0, 1\rangle$ to $\langle user, 1\rangle$).

We make two design choices here. The first is to associate the service instance concept as a property (label) of a condition (e.g., $\langle http, 1\rangle$), instead of an exploit (as in our previous work [6]). This label can then be inherited by the corresponding exploits. The second design choice is that, while some conditions indicate the involved firewall rules, the actual label values that they will take will depend on the number of predefined modifiable rules in the firewall itself. For each firewall, instead of modeling service instances,

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

we model the number of modifiable firewall rules that can be enabled. This would help to avoid the need for introducing new conditions and exploits into the extended resource graph when firewall rules are to be disabled and hence we may work with a fixed structure of the extended resource graph. While the definitions of service pool and service instance remain the same as in [6], Definitions 1 and 2 formally introduce the revised concepts.

Definition 1 (Firewall Rule Pool and Firewall Rule). *Denote F the set of all firewalls and Z the set of integers, for each firewall $f \in F$, the function $r(.) : F \to \mathsf{Z}$ gives the firewall rule pool of f which represent all modifiable firewall rules of that firewall.*

Definition 2 (Extended Resource Graph). *Given a network composed of*

- *a set of hosts H,*
- *a set of services S, with the service mapping $serv(.) : H \to 2^S$,*
- *the collection of service pools $SP = \{sp(s) \mid s \in S\}$,*
- *the collection of firewall rules $FR = \{r(f) \mid f \in F\}$,*
- *a set of firewalls F, with the rule mapping $r(.) : F \to \mid FR \mid$,*
- *and the labelling function $v(.) = v_f(.) \cup v_c(.)$ where $v_f(.) : f \to F$ and $v_c(.) : C \to SP$*

Let E be the set of zero day exploits $\{\langle s, h_s, h_d\rangle \mid h_s \in H, h_d \in H, s \in serv(h_d)\}$, and $R_r \subseteq C \times E$ and $R_i \subseteq E \times C$ be the collection of pre and post-conditions in C. We call the labeled directed graph, $\langle G(E \cup C, R_r \cup R_i), v\rangle$ the extended resource graph.

2.2 Heterogeneous Hardening Control and Cost Model

We introduce the notion of *heterogeneous hardening control* as a model to account for all hardening options in a network where we represent each initial condition as an optimization variable. We formulate the heterogeneous hardening control vectors using those variables as follows. We note that the number of optimization variables present in a network will depend on the number of initial conditions that are affected by one or more hardening options. Since we only consider remotely accessible services in the extended resource graph model, we would expect in practice the number of optimization variables to grow linearly in the size of the network (i.e., the number of hosts).

Definition 3 (Optimization Variable and Heteregeneous Hardening Control). *Given an extended resource graph $\langle G, v\rangle$, $\forall c \in C$ and $\forall f \in F$, $v(c)$ and $v(f)$ are optimization variables. A hardening control vector is the integer valued vector $\boldsymbol{V} = (v(c_1), v(c_2), ..., v(c_{|C|}) \cup (v(f_1), v(f_2), ..., v(f_{|F|})$*

Changing the value of an optimization variable has an associated *hardening cost* and the collection of such costs are given in a *hardening cost matrix* in a self-explanatory manner. We make use of Gartner's 2003 *Total Cost of Ownership* (TCO) analysis report [20] to establish a realistic cost estimation of the cost of different hardening options. Table 1 provides a reference as to which criteria is applicable to different hardening options costs.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Hardening Option Cost Selection Criteria				
Gartner's TCO criteria	Firewall Connectivity	Firewall Layer 3	Firewall Access Control	Diversity
Downtime Costs				x
Operational Costs	x	x	x	x
Support Costs	x		x	x
Changes in upgrade Costs	x	x	x	x
Monitoring costs			x	x
Production costs				x
Security management and failure control costs	x	x	x	x

Table 1. Criteria to be used when calculating hardening costs for different hardening options based on Gartner's TCO [20]

Definition 4 (Hardening Cost). *Given $s \in S$ and $sp(s)$, and given $f \in F$ and $r(f)$, the cost to change from one specific hardening option to another is defined as the hardening cost.*

Definition 5 (Hardening Cost Matrix). *The collection of all hardening costs for all hardening options are given as a hardening cost matrix HCM. For the different hardening options, the element at i^{th} row and j^{th} column indicates the hardening cost of changing the i^{th} hardening option to be the j^{th} hardening option.*

Definition 6 (Total Hardening Cost). *Let $v_s(c_i)$ be the service associated with the optimization variable $v(c_i)$ and $\boldsymbol{V}_{c0}$ the initial service instance values for each of the conditions in the network. Let $v_f(f_i)$ be the firewall associated with the optimization variable $v(f_i)$ and $\boldsymbol{V}_{f0}$ the initial firewall rule set values for each of the firewalls in the network. The total hardening cost, Q_d, given by the heterogeneous hardening vector $\boldsymbol{V}$ is obtained by*

$$Q_d = \sum_{i=1}^{|C|} CM_{v_s(c_i)}(\boldsymbol{V}_{c0}(i), \boldsymbol{V_c}(i)) + \sum_{i=1}^{|F|} CM_{v_f(f_i)}(\boldsymbol{V}_{f0}(i), \boldsymbol{V_f}(i))$$

We note that the above definition of hardening cost between each pair of service instances has some advantages. For example, in practice we can easily imagine cases where the cost is not symmetric, i.e., changing one service instance to another (e.g., from Apache to IIS) carries a cost that is not necessarily the same as the cost of changing it back (from IIS to Apache). Our approach of using a collection of two-dimensional matrices allows us to account for cases like this. Additionally, by considering instance 0, it provides us the advantage to model disabling a service as a special case of service diversification, if the hardening option allows it.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2.3 Problem formulation

As mentioned in Section 1, the security metric that we will be using, denoted as d, is based on the minimum number of distinct resources, excluding those with unpatchable vulnerabilities, on the shortest attack path in the resource graph [26], with the extension for considering the uneven distribution of services along that path [30, 33], as formally defined below.

Definition 7 (d-Safety Metric). *Given an extended resource graph $\langle G(E \cup C, R_r \cup R_i), v\rangle$, and a goal condition $c_g \in C$; let $t = \sum_{i=1}^{n} 2^{-n} \mid serv(h_i)) \mid$ (total number of service instances), and let $p_j = \frac{|h_i : s_j \in serv(h_i))|}{t}$ ($1 \leq i \leq n$, $1 \leq j \leq n$) (relative frequency of each resource). For each $c \in C$ and $q \in seq(c)$ (attack path), denote $R(q)$ for $s : s \in$ R, r appears in q, r is not unpatchable, we define the network's d-safety metric (where min(.) returns the minimum value in a set) $d = min_{q \in seq(c_g)} r(R(q))$; where $r(R(q))$ is the attack path's effective richness of the services, defined as $r(G) = \frac{1}{\prod_1^n p_i^{p_i}}$ [30]*

With the aforementioned models, the network hardening problem is to maximize the d value by changing the hardening options while respecting the available budget in terms of given cost constraints. In the following, we formally formulate this as an optimization problem.

Problem 1 (d-Optimization Problem). Given an extended resource graph $\langle G, v\rangle$, find a heterogeneous hardening control vector $\boldsymbol{V}$ which maximizes $min(d(\langle G(\boldsymbol{V}), v\rangle))$ subject to the constraint $Q \leq B$, where B is the available budget and Q is the total hardening cost as given in Definition 6.

Since our problem formulation is based on an extended version of the resource graph, which is syntactically equivalent to attack graphs, many existing tools developed for the latter (e.g., the tool in [16] has seen many real applications to enterprise networks) may be easily extended to generate extended resource graphs which we need as inputs. Additionally, our problem formulation assumes a very general model of budget B and cost Q, which allows us to account for different types of budgets and cost constraints that an administrator might encounter in practice, as will be demonstrated in the following section.

3 The Methodology

This section details our optimization and heuristic algorithms used for solving the formulated heterogeneous hardening problem. We also illustrate the optimization process through a few case studies.

3.1 Optimization Algorithm

Our first task is to select an optimization algorithm that would fit our hardening problem. First, it is well known that most gradient-based methods require to satisfy mathematical properties like convexity or differentiability, which are not applicable to our

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

problem. Second, the problem we want to solve includes different if-then-else constructs to account for the different hardening technique used, and thus, an algorithm that allows to insert this construct is necessary. Additionally, since our optimization problem uses variables that are defined as discrete (discrete variable space), a simple and robust search method and optimization technique is needed. We find that metaheuristic algorithms provide these advantages. Specifically, the Genetic Algorithm (GA) provides a simple and clever way to encode candidate solutions to the problem [8]. One of the main advantages is that we do not have to worry about explicit mathematical definitions. For our automated optimization approach, we chose GA because it requires little information to search effectively in a large search space in contrast to other optimization methods (e.g., the mixed integer programming [4]).

The extended resource graph is the input to our automated optimization algorithm where the fitness function is d. One important point to consider when optimizing the d function on the extended resource graph is that, for each generation of the GA, the graph's labels selected will dynamically change. This in turn will change the value of d, since the shortest path may have changed with each successive generation of GA and the change in the hardening options will enable or disable certain paths. Our optimization tool takes this into consideration. Additionally, if there are more than one shortest path that provides the optimized d, our optimization tool gives priority to the paths by considering the uneven distribution and relative frequency of resources in that path, thus addressing one of the limitations that was present in [6] where no priority was provided.

The constraints are defined as a set of inequalities in the form of $q \leq b$, where q represents one or more constraint conditions and b represents one or more budgets. These constraint conditions can be overall constraints (e.g., the total hardening cost Q_d) or specific constraints to address certain requirements or priorities while implementing the heterogeneous hardening options. The number of independent variables used by the GA (genes) are the optimization variables given by the extended resource graph. For our particular network hardening problem, the GA will be dealing with integer variables representing the selection of a hardening option. Because $v(.)$ is defined as an integer, the optimization variables need to be given a minimum value and a maximum value. This range is determined by the number of instances provided in the service pool of each service and firewall rule pool of each firewall. The initial service instance for each of the services and the initial set of firewall rules are given by the extended resource graph while the final heterogeneous hardening control vector $\boldsymbol{V}$ is obtained after running the GA.

3.2 Case Studies

In the following, we demonstrate different use cases of our method with varying cost constraints and hardening options. For these test cases, the population size defined for our tool is set to be at least the value of optimization variables (more details will be provided in the coming section). This way we ensure the individuals in each population span the search space. We ensure the population diversity by testing with different settings in genetic operations (like crossover and mutation). For all the test cases, we have used the following algorithm parameters: population size = 100, number of generations = 150, crossover probability = 0.8, and mutation probability = 0.2.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

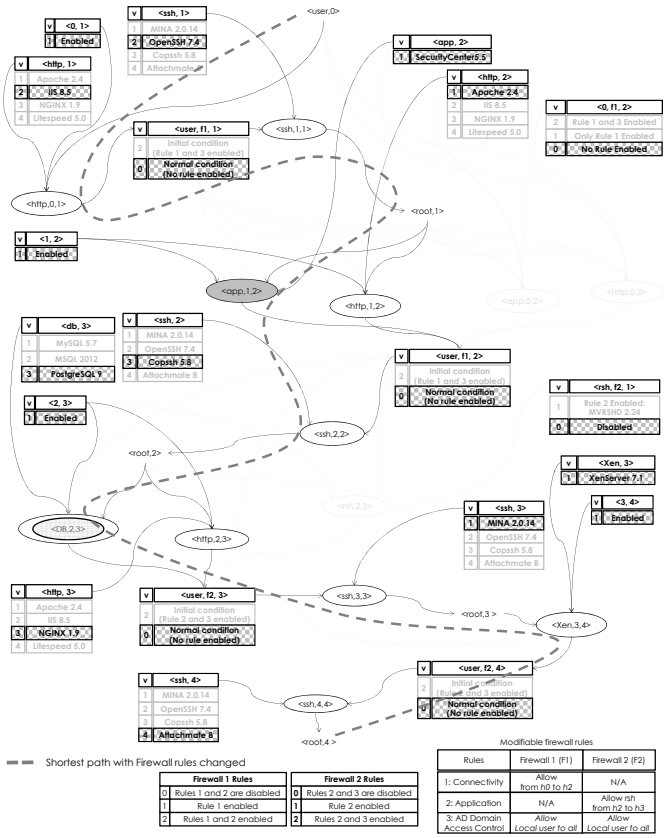


Fig. 3. Test case A: Effect of modifiable hardening options and budget constraints.

Test case A: $Q_d \leq 500$ units with firewall rule constraints. We start with the simple case of one overall budget constraint ($Q_d \leq 500$). There are 11 different services-based optimization variables and 2 firewall-based optimization variables. If no firewall rules are changed, the solution provided by the GA yields d=2.7529. In this case, because of the firewall rules that are enabled, the metric cannot be increased any further.

On the other hand, if we allow the firewall rules to be modified, while maintaining the overall budget $Q_d \leq 500$, the optimization results will be quite different. The solution provided by the GA is a d metric of 3.3895. This total hardening cost satisfies both the overall budget constraints. We can see that the hardening options enforced by the firewall rules in our optimization tool can affect the optimization. Nevertheless, additional budget constraints might not allow achieving the maximum d possible.

Test case B: $Q_d \leq 500$ units with a critical service with an unpatched vulnerability. While test case A shows how enabling or disabling predefined firewall rules can affect the d metric optimization, when considering the effects of unpatchable vulnerabilities the d metric value will change. This test case models such a scenario by assigning a restriction for the ssh services not to be diversified or disabled.

In the graph, we can see that the ssh service is highlighted to represent the fact that it cannot be patched. The solution provided by the GA is d=2.8284. While the increase

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

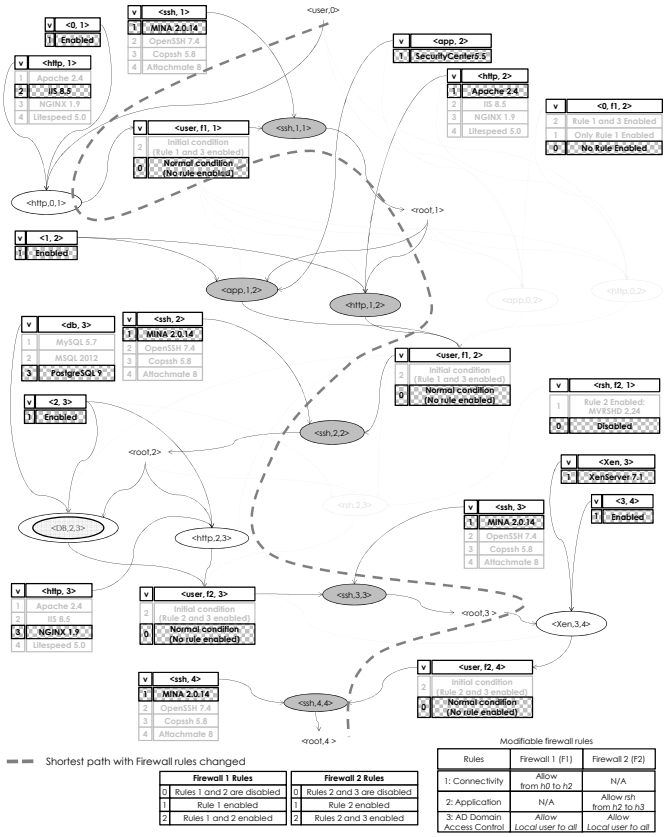


Fig. 4. Test case B: Effect of having an unpatchable vulnerability in the network.

is less than when the ssh service can be diversified, we can still have an increase in the d metric even with unpatchable vulnerabilities on the network.

As seen from the above test cases, our model and problem formulation makes it relatively straightforward to apply any standard optimization techniques, such as the GA, to optimize the d metric through combining different network hardening options while dealing with unpatchable and unknown vulnerabilities and respecting given cost constraints.

3.3 Heuristic Algorithm

All the test cases described above rely on the assumption that all the attack paths are readily available. However, this is not always the case in practice. Due to the well-known complexity that resource graphs have inherited from attack graphs due to their common syntax [30, 33], it is usually computationally infeasible to enumerate all the available attack paths in a resource graph for large networks. Therefore, we present a modified version of the heuristic algorithm [6] to reduce the search complexity when calculating and optimizing the d metric by only storing the m-shortest paths at each step. The following briefly describes the modified algorithm.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The algorithm starts by finding the initial conditions that are affected by the modifiable firewall rules and stores them on a list γ. After that, it topologically sorts the graph and proceeds to go through each one of the nodes on the resource graph. If an exploit is a post-condition of one of the conditions in γ, it is not included in the set of exploits $\sigma()$. The main loop cycles through each unprocessed node. If a node is an initial conditions, the algorithm assumes that the node itself is the only path to it and it marks it as processed. For each exploit e, all of its preconditions are placed in a set. The collection of attack paths $\alpha(e)$ is constructed from the attack paths of those preconditions. In a similar way, $\sigma'(ov(e))$ is constructed with the function $ov()$ which, aside from using the exploits, includes value of elements of the hardening control vector that supervises that exploit.

If there are more than m paths to that node, the algorithm will first look for the relative frequency of each unique combination of exploit and service instance in $\alpha'(ov(e))$. Then, the algorithm creates a dictionary structure where the key is a path from $\alpha(e)$ and the value is the effective richness of service/service instance combinations given by each one of the respective paths in $\alpha'(ov(e))$. A function $ShortestM()$ selects the top m keys whose values are the smallest and returns the m paths with the smallest effective richness value. If there are less than m paths, it will return all of the paths. After this, it marks the node as processed. The process is similar when going through each one of the intermediate conditions. Finally, the algorithm returns the collection of m paths that can reach the goal condition c_g. It is worth noting that by considering the effective richness of each path, the algorithm provides gives a path a priority based on the relative frequency of the combination of unique service with service instance.

4 Simulations

In this section, we show simulation results. All simulations are performed using a computer equipped with a 3.0 GHz CPU and 8GB RAM in the Python 2.7.10 environment under Ubuntu 12.04 LTS and MATLAB 2015a's GA toolbox. To generate a large number of resource graphs for simulations, we first construct a small number of seed graphs based on realistic cloud networks and then generate larger graphs from those seed graphs by injecting new hosts and assigning resources in a random but realistic fashion (e.g., the number of pre-conditions of each exploit is varied within a small range since real world exploits usually have a constant number of pre-conditions).

For the different hardening options that are implemented through firewall rules, we randomly select 10% of the initial conditions. Additionally, to analyze the effect of unpatchable vulnerabilities, our graphs include randomly assigned unpatchable services. The resource graphs are used as the input for the optimization toolbox where the objective function is to maximize the minimum d value subject to budget constraints. In all the simulations, we employ the heuristic algorithm described in section 3.3.

To determine the genetic operators, we used the hill climbing algorithm. Our simulations showed that (detailed simulation results are omitted here due to page limitations), using the GA with a crossover probability of 80%, a mutation rate of 20%, and setting the number of generations to 70 will be sufficient to obtain good results. Additionally, our experiences also show that, because our largest resource graph had a heterogeneous

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

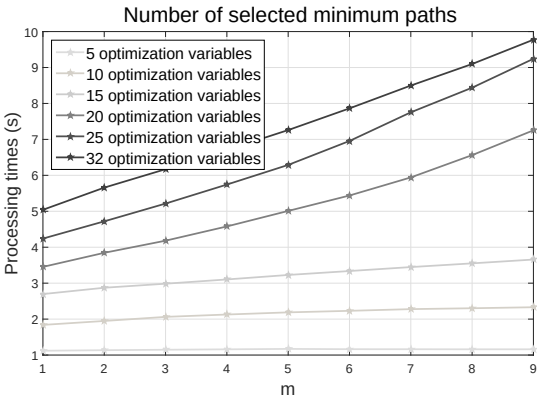


Fig. 5. Processing time.

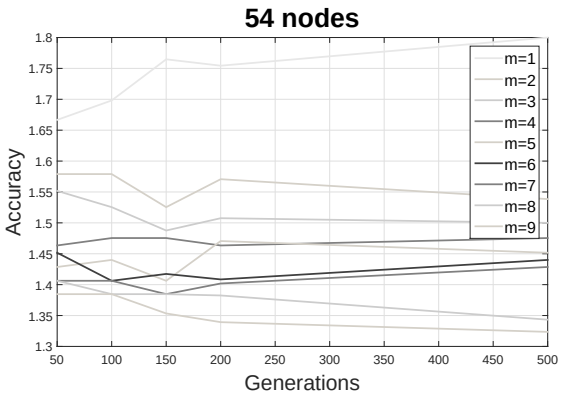


Fig. 6. Accuracy vs m (parameter of the heuristic algorithm).

hardening control vector of less than 100 variables, we could set the population size equal to 200; nevertheless, we believe that when dealing with a bigger number of optimization variables, the population size should be at least twice the number of variables.

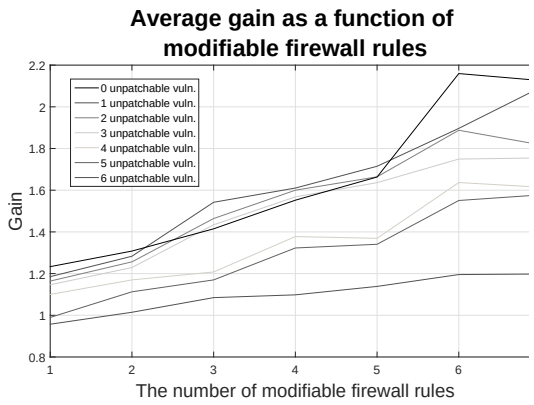


Fig. 7. The average gain based on the number of modifiable firewall rules.

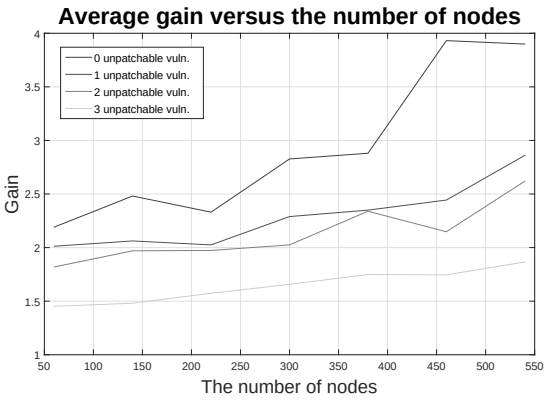


Fig. 8. The average gain vs the number of nodes.

Figure 5 shows that the processing time increases almost linearly as we increase the number of optimization variables or the parameter m of the heuristic algorithm. The results show that the algorithm is relatively scalable with a linear processing time. On the other hand, the accuracy of the results is also an important issue to be considered. Here the accuracy refers to the approximation ratio between the result obtained using the heuristic algorithm and that of the brute force algorithm (i.e., simply enumerating and searching all the paths while assuming all services and service instances are dif-

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

ferent). For the simulations depicted in Figure 6, we settled for 50 iterations per graph per m-paths. The heterogeneous hardening control vector provided by the GA is used to calculate the accuracy. From the results, we can see that when m is greater or equal to 4 the approximation ratio reaches an acceptable level. For the following simulations, we have settled with an m value of 6.

Figure 7 shows that the gain will increase linearly as we increase the number of firewall-based hardening options. These results confirm that firewall-based hardening options can positively affect our effort to provide better resilience for cloud networks against zero-day attacks. Additionally, the figure shows that the number of unpatchable vulnerabilities that are present in the network will significantly reduce the gain that can be achieved through other hardening techniques. Since it is not probable to find a large number of unpatchable vulnerabilities all at the same time within a network, we only consider up to three unpatchable vulnerabilities.

In Figure 8, we analyze the average gain in the optimized results for different sizes of graphs. In this figure, we can see that we have a good enough gain for graphs with a relatively high number of nodes. As expected, as we increase the number of unpatchable vulnerabilities, the gain will decrease. However, we can also see this decrease is linear. For those simulations, we have used a population size of 300, 50 generations, and a crossover fraction of 80%.

Figures 9 and 10 show the optimization results of different shapes of resource graphs in terms of depth and degree of exposure, which roughly represents the extent to which the network is protected. While it may be difficult to exactly define the depth of a resource graph, we have relied on the relative distance, i.e., the difference of the shortest path before and after all hardening options have been applied. There is a linear increase in the gain as we increase the relative distance in the shortest path. This is independent of the amount of unpatchable vulnerabilities. While this does not provide an accurate description of the graph's shape, it does provide an idea of how much our algorithm can increase the minimum d for graphs with different depths.

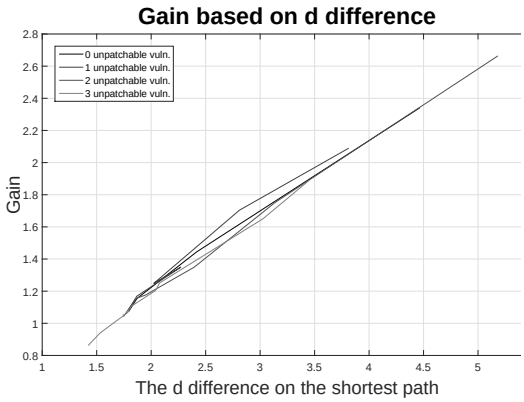


Fig. 9. The d difference on the shortest path.

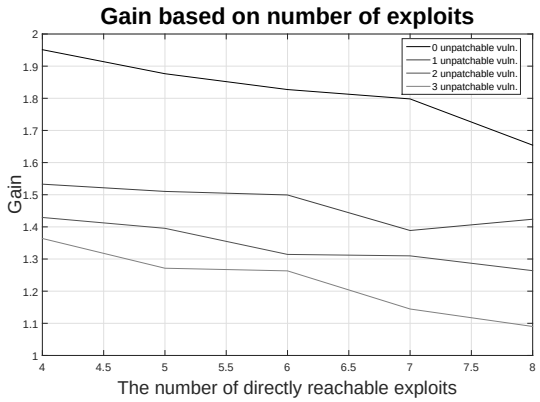


Fig. 10. The number of directly reachable exploits.

Finally, as shown in Figure 10, we can see the effect of the network's degree of exposure, which is defined as the number of exploits that are directly reachable by the attacker from the external host *h0*. As we increase the degree of exposure, the gain in optimization decreases (circles in the graph). That is, there will less room for hardening if the network is more exposed.

5 Related Work

In general, the security of networks may be qualitatively modeled using attack trees [9, 10, 23] or attack graphs [2, 24]. A majority of existing quantitative models of network security focus on known attacks [29, 1], while few works have tackled zero day attacks [27, 26, 30, 33] which are usually considered unmeasurable due to the uncertainties involved [18]. In terms of security metrics, most of the current works deal with assigning numeric scores to rank known vulnerabilities (mostly based on the CVSS) [19] to be able to model the impact that they have on a network. This ranking is based on how likely and easily exploitable the known vulnerabilities are. This, however, is not the case for unknown vulnerabilities.

Early works on network hardening typically rely on qualitative models while improving the security of a network [24, 28, 25]. Those works secure a network by breaking all the attack paths that an attacker can follow to compromise an asset, either in the middle of the paths or at the beginning (disabling initial conditions). Also, those works do not consider the implications when dealing with budget constraints nor include cost assignments, and tend to leave that as a separate task for the network administrators. While more recent works [1, 32] generally provide a cost model to deal with budget constraints, one of the first attempts to systematically address this issue is by Gupta et al. [15]. The authors employed genetic algorithms to solve the problem of choosing the best set of security hardening options while reducing costs.

Dewri et a. [9] build on top of Gupta's work to address the network hardening problem using a more systematic approach. They start by analyzing the problem as a single objective optimization problem and then consider multiple objectives at the same time. Their work consider the damage of compromising any node in the cost model in order to determine the most cost-effective hardening solution. Later on, in [10] and in [31], the authors extrapolate the network hardening optimization problem as vulnerability analysis with the cost/benefit assessment, and risk assessment respectively. In [22] Poolsappasit et al. extend Dewri's model to also take into account dynamic conditions (conditions that may change or emerge while the model is running) by using Bayesian attack graphs in order to consider the likelihood of an attack. Unlike our work, most existing work is limited to known vulnerabilities and focus on disabling existing services.

There exist a rich literature on employing diversity for security purposes. The idea of using design diversity for tolerating faults has been investigated for a long time, such as the N-version programming approach [3], and similar ideas have been employed for preventing security attacks, such as the N-Variant system [7], and the behavioral distance approach [13]. In addition to design diversity and generated diversity, recent work employ opportunistic diversity which already exists among different software systems. For example, the practicality of employing OS diversity for intrusion tolerance is

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

evaluated in [14]. More recently, the authors in [30, 33] adapted biodiversity metrics to networks and lift the diversity metrics to the network level. While those works on diversity provide motivation and useful models, they do not directly provide a systematic solution for improving diversity. So far, the work done by [6], is one of the first work that has tried to provide a solution for this problem; their limitation, however, is that their metric is too simplistic and does not consider additional hardening options.

6 Conclusions

In this paper, we have provided a heterogeneous approach to network hardening to increase the resilience of a network against both unknown and unpatchable vulnerabilities. By unifying different hardening options within the same model, we derived a more general method than most existing efforts that rely on a single hardening option. Our automated approach employed a heuristic algorithm that helped to manage the complexity of evaluating the security metric as well as limiting the time for optimization to an acceptable level. We have addressed one limitation of our previous work by considering the uneven distribution of services along an attack path. We have devised a more realistic cost model. We have tested the efficiency and accuracy of the proposed algorithms through simulation results, and we have also discussed how the gain in the d value will be affected by the number of available modifiable firewall rules, unpatchable vulnerabilities, and the different sizes and shapes of the resource graphs.

The following lists several future direction of our approach.

- While this paper has proven that we can integrate different network hardening options (e.g., firewalls and diversity) under the same model, some hardening options may not easily fit into this model (e.g., service relocation).
- The security metric we applied relies on the number of unknown vulnerabilities, which may be refined by further considering known and patchable vulnerabilities (even though those would carry less weight).
- This study relies on a static network configuration. A future research direction would be to consider a dynamic network model in which both attackers and defenders may cause incremental changes in the network.
- We will evaluate other optimization algorithms in addition to GA to find the most efficient solution for our problem.

Disclaimer Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

1. Massimiliano Albanese, Sushil Jajodia, and Steven Noel. Time-efficient and cost-effective network hardening using attack graphs. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*, pages 1–12. IEEE, 2012.

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2. Paul Ammann, Duminda Wijesekera, and Saket Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 217–224. ACM, 2002.
3. Algirdas Avizienis and Liming Chen. On the implementation of n-version programming for software fault tolerance during execution. In *Proc. IEEE COMPSAC*, volume 77, pages 149–155, 1977.
4. H Md Azamathulla, Fu-Chun Wu, Aminuddin Ab Ghani, Sandeep M Narulkar, Nor Azazi Zakaria, and Chun Kiat Chang. Comparison between genetic algorithm and linear programming approach for real time operation. *Journal of Hydro-environment Research*, 2(3):172–181, 2008.
5. Kapil Bakshi. Cisco cloud computing-data center strategy, architecture, and solutions. *CISCO White Paper. Retrieved October*, 13:2010, 2009.
6. Daniel Borbor, Lingyu Wang, Sushil Jajodia, and Anoop Singhal. Diversifying network services under cost constraints for better resilience against unknown attacks. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 295–312. Springer, 2016.
7. Benjamin Cox, David Evans, Adrian Filipi, Jonathan Rowanhill, Wei Hu, Jack Davidson, John Knight, Anh Nguyen-Tuong, and Jason Hiser. N-variant systems: a secretless framework for security through diversity. In *Usenix Security*, volume 6, pages 105–120, 2006.
8. Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2):311–338, 2000.
9. Rinku Dewri, Nayot Poolsappasit, Indrajit Ray, and Darrell Whitley. Optimal security hardening using multi-objective optimization on attack tree models of networks. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 204–213. ACM, 2007.
10. Rinku Dewri, Indrajit Ray, Nayot Poolsappasit, and Darrell Whitley. Optimal security hardening on attack tree models of networks: a cost-benefit analysis. *International Journal of Information Security*, 11(3):167–188, 2012.
11. Wayne W Eckerson. Three tier client/server architectures: achieving scalability, performance, and efficiency in client/server applications. *Open Information Systems*, 3(20):46–50, 1995.
12. Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian. *OpenStack Operations Guide*. " O'Reilly Media, Inc.", 2014.
13. Debin Gao, Michael K Reiter, and Dawn Song. Behavioral distance measurement using hidden markov models. In *Recent Advances in Intrusion Detection*, pages 19–40. Springer, 2006.
14. Miguel Garcia, Alysson Bessani, Ilir Gashi, Nuno Neves, and Rafael Obelheiro. Os diversity for intrusion tolerance: Myth or reality? In *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*, pages 383–394. IEEE, 2011.
15. Mukul Gupta, Jackie Rees, Alok Chaturvedi, and Jie Chi. Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach. *Decision Support Systems*, 41(3):592–603, 2006.
16. S. Jajodia, S. Noel, and B. O'Berry. Topological analysis of network attack vulnerability. In V. Kumar, J. Srivastava, and A. Lazarevic, editors, *Managing Cyber Threats: Issues, Approaches and Challenges*. Kluwer Academic Publisher, 2003.
17. B. Krebs. How many zero-days hit you today? http://krebsonsecurity.com/2013/12/how-many-zero-days-hit-you-today/, 2013.
18. John McHugh. Quality of protection: measuring the unmeasurable? In *Proceedings of the 2nd ACM workshop on Quality of protection*, pages 1–2. ACM, 2006.
19. Peter Mell, Karen Scarfone, and Sasha Romanosky. Common vulnerability scoring system. *Security & Privacy, IEEE*, 4(6):85–89, 2006.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

20. Lars Mieritz and Bill Kirwin. Defining gartner total cost of ownership. *L. Mieritz, B. Kirwin*, 2005.
21. Apache mina project. https://mina.apache.org/mina-project/, Oct, 2016.
22. Nayot Poolsappasit, Rinku Dewri, and Indrajit Ray. Dynamic security risk management using bayesian attack graphs. *Dependable and Secure Computing, IEEE Transactions on*, 9(1):61–74, 2012.
23. Indrajit Ray and Nayot Poolsapassit. Using attack trees to identify malicious attacks from authorized insiders. In *ESORICS 2005*, pages 231–246. Springer, 2005.
24. Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, and Jeannette M Wing. Automated generation and analysis of attack graphs. In *Security and privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 273–284. IEEE, 2002.
25. Lingyu Wang, Massimiliano Albanese, and Sushil Jajodia. *Network Hardening: An Automated Approach to Improving Network Security*. Springer Publishing Company, Incorporated, 2014.
26. Lingyu Wang, Sushil Jajodia, Anoop Singhal, Pengsu Cheng, and Steven Noel. k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *Dependable and Secure Computing, IEEE Transactions on*, 11(1):30–44, 2014.
27. Lingyu Wang, Sushil Jajodia, Anoop Singhal, and Steven Noel. k-zero day safety: Measuring the security risk of networks against unknown attacks. In *ESORICS 2010*, pages 573–587. Springer, 2010.
28. Lingyu Wang, Steven Noel, and Sushil Jajodia. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18):3812–3824, 2006.
29. Lingyu Wang, Anoop Singhal, and Sushil Jajodia. Measuring the overall security of network configurations using attack graphs. In *Data and Applications Security XXI*, pages 98–112. Springer, 2007.
30. Lingyu Wang, Mengyuan Zhang, Sushil Jajodia, Anoop Singhal, and Massimiliano Albanese. Modeling network diversity for evaluating the robustness of networks against zero-day attacks. In *ESORICS 2014*, pages 494–511. Springer, 2014.
31. Shuzhen Wang, Zonghua Zhang, and Youki Kadobayashi. Exploring attack graph for cost-benefit security hardening: A probabilistic approach. *Computers & security*, 32:158–169, 2013.
32. Beytullah Yigit, Gurkan Gur, and Fatih Alagoz. Cost-aware network hardening with limited budget using compact attack graphs. In *Military Communications Conference (MILCOM), 2014 IEEE*, pages 152–157. IEEE, 2014.
33. M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese. Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks. *IEEE Transactions on Information Forensics and Security (TIFS)*, 11(5):1071–1086, 2016.
34. Mengyuan Zhang, Lingyu Wang, Sushil Jajodia, Anoop Singhal, and Massimiliano Albanese. Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks. *IEEE Transactions on Information Forensics and Security*, 11(5):1071–1086, 2016.

Borbor, Daniel; Jajodia, Sushil; Singhal, Anoop; Wang, Lingyu.
"Securing Networks Against Unpatchable and Unknown Vulnerabilities Using Heterogeneous Hardening Options."
Paper presented at 31st IFIP Conference on Data and Application Security and Privacy (DBSEC 2017),
Philadelphia, PA. July 19, 2017 - July 21, 2017.

Real-Time Scheduling for Wireless Networks with Random Deadlines

Mohamed Kashef, *Member, IEEE* and Nader Moayeri, *Senior Member, IEEE*

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

***Abstract*—The use of wireless communications in industrial environments is motivated by the flexibility that wireless networks provide and their cost-efficient setup and maintenance. Various wireless technologies have been introduced to satisfy the strict industrial requirements. Time division multiple access (TDMA) protocols have been widely exploited in various wireless technologies due to the ease of implementation and packets collision avoidance. In this work, we consider the problem of scheduling multiple flows over a wireless network in industrial environments. These flows represent the data coming from the sensors to the controller and the control commands going to the actuators from the controllers. These flows are characterized by random strict deadlines for each packet in a flow following a given probability distribution. Moreover, the schedule is built over a frame of transmissions with the objective of minimizing the total number of packets missing their deadlines. We obtain the optimal scheduling scheme by formulating and solving an unobservable Markov decision problem (UMDP). Then, we obtain a sub-optimal scheduling scheme which has a near-optimal performance for a wide range of system parameters. Finally, we evaluate these scheduling schemes numerically to study the effects of various system parameters on the performance.**

I. Introduction

Various networking protocols have been introduced to meet the requirements of the industrial enviornments and improve control capabilities. Typically, wired networks have been employed based on structured cabling which is defined as cabling infrastructure that consists of a number of standardized smaller elements. Lately, industrial wireless technologies have been recognized as attractive alternatives to their wired counterparts due to increased networking flexibility, and lower setup and maintenance costs. These wireless technologies include WirelessHART and ISA 100.11a. Various advantages and disadvantages of these technologies have been considered in the literature [1], [2].

The main challenge facing deployment of wireless networks in industrial environments is the lack of reliability of such networks. The nature of wireless channel leads to considerable bit errors and hence packets can be lost or delayed past their intended delivery deadlines. Due to the importance of timing, strict deadlines are commonly enforced for data packets in industrial environments. To achieve these reliability goals, time division multiple access (TDMA)-based medium access control (MAC) protocols are used to eliminate the possibility of intra-network interference and to increase the likelihood that packets would meet their deadline requirements.

One of the major approaches to enhance wireless network reliability is scheduling which ensures that only one network node transmits at any given time. In addition, scheduling makes it possible to introduce redundancy by transmitting the same information packet over several time slots to increase the chance of the packet reaching its destination correctly and before its deadline. Many survey papers have been written on scheduling in wireless sensor networks (WSN). The work in [3] describes the network parameters that affect TDMA scheduling and considers various performance measures including latency, synchronization time, and energy consumption, and the communication patterns supported by various algorithms. Moreover, heuristic scheduling algorithms have been surveyed in [4]. The scheduling algorithms that use guaranteed time slot (GTS) in IEEE 802.15.4-based networks have been compared in [5].

In this work, we consider the TDMA-based scheduling problem for multiple real-time flows with strict deadlines. The goal of the schedule is to minimize the total number of packets missing their deadlines. The schedule is built at the beginning of a frame to reduce the computational complexity of decision making at every time slot. It is then updated at the beginning of every new frame. The routing protocol is assumed to be known. Therefore, the route for any flow from a certain source to a certain destination is predetermined before the scheduling problem is solved. The data flows carry either sensing data or control commands. Hence, the packet generation process and the deadlines are not assumed to be periodic. Instead, they are assumed to have a stochastic nature with known probability density functions.

M. Kashef and N. Moayeri are with Advanced Networking Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA 20899 (e-mail: {mohamed.hany,nader.moayeri}@nist.gov).

U.S. Government work not protected by U.S. copyright

Hany, Mohamed; Moayeri, Nader.
"Real-Time Scheduling for Wireless Networks with Random Deadlines."
Paper presented at 13th IEEE International Workshop on Factory Communication Systems, Trondheim, Norway. May 31, 2017 - June 2, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Sensing data flows are typically periodic, but there are also event-based signals that are aperiodic. The same is true for control commands. This has motivated the consideration of data flows with random deadlines in this paper. The problem of scheduling flows with random deadline has been considered in the literature [6]-[9]. In [6], a multi-class queuening network is considered where the customers arrive in the network with random deadlines. Each class has its own deadline probability distribution. In [7], the problem of online packet scheduling is considered in a multi-hop wired network, where each packet has a weight indicating the importance of being delivered by the deadline and the cumulative weights of packets delivered before their deadlines are maximized using admission control and scheduling. In [8], a packet scheduling algorithm is proposed to optimize the performance in the case of having both real-time and best effort flows concurrently. The traffic characteristics of the flows are considered to be stochastic. The randomness of deadlines in these works, namely [6]-[8], may arise from the initial queueing delays and the randomnes of data processing. In [9], the problem of providing timeliness guarantees for multi-hop messages is considered, where the effective deadlines of messages are determined by how long the data is valid/operative and by what time it is needed at the destination. The validity of the data depends on the underlying physical processes and hence can be random and the time it is needed at the destination is determined by the criticality of the data and the time it takes to process the data. In this paper, we solve the scheduling problem for frame-based transmissions in a multi-hop network with random deadlines where each wireless link has a certain packet transmission success probability.

The problem of minimizing the number of packets missing their deadlines is formulated as an unobservable Markov decision problem (UMDP), which is a special case of partially observable Markov decision problems (POMDPs) [10]. The system state is not observable as we compute the schedule at the beginning of the frame. The system state is defined by the numbers of remaining time slots before the deadlines and the numbers of remaining hops till the destinations. The complexity of the obtained algorithm using the solution of the UMDP has motivated finding a sub-optimal solution for the problem through modifying the commonly used earliest deadline first (EDF) algorithm to consider the randomness in deadlines.

The rest of this paper is organized as follows. The system model is presented in Section II. The optimal scheduling problem is formulated and solved in Section III. The sub-optimal scheduling is discussed in Section IV. In Section V, numerical results are presented. Finally, Section VI presents the concluding remarks.

II. System Model

Consider a set of M wireless flows $\mathcal{F} = \{F_1, F_2, ...F_M\}$ to be scheduled over a single frequency band. The route for F_m is denoted by ϕ_m. The length of ϕ_m, denoted by h_m^*, is the total number of hops a packet in F_m has to go through before it gets to its destination. A new packet arrives in F_m when the deadline for the preceding packet in the same flow expires. The schedule for the network flows is calculated every hyper-period T, which consists of a fixed number of time slots. In the case of an industrial environment with field devices that generate packets in a periodic manner and the packet generation period is a multiple of the duration of a time slot, the hyper-period is commonly defined in the literature as the least common multiple of the packet generation periods of the field devices [11].

This paper does not consider the possibility of spatial frequency reuse. Therefore, at most one node in the network transmits in each time slot. This assumption is inline with how WirelessHART and ISA100.11a are used. The wireless channel between a node i and a node j, representing a single hop in the route of F_m, is modeled as a binary erasure channel that is independent of the packet generation process. The quality of this channel is represented by the probability $\rho_{i,j}$ of successful transmission of a packet, called (packet) success probability. The success probability is determined by the channel impulse response for the physical channel between nodes i and j as well as the wireless communications system's physical parameters such as transmission power, modulation scheme, coding scheme, and targeted bit-error rate. Even though the scheduling algorithms developed in this paper can handle the possibility of having different success probabilities for the channels between different pairs of nodes, we assume that $\rho_{i,j} = \rho$, for all (i,j), for the sake of simplicity.

Each packet in F_m has a deadline that is modeled by a positive discrete random variable D_m with probability mass function $f_m(.)$. The discrete random variable D_m takes positive integer values denoted by $d_m \in \mathcal{B}_m$ where $\mathcal{B}_m$ is the space of the random deadlines and it can be be any subset of $\{h_m^*, h_m^* + 1, \ldots, D_m^*\}$, where D_m^* can be infinite. The mean and variance of D_m are denoted by μ_m and σ_m^2, respectively. We consider data transmission with strict deadlines, such that each packet in F_m is discarded if it has not reached its intended destination prior to its deadline. Regardless of whether a packet in F_m succeeds or fails to reach its destination by its

deadline, the next packet in this flow is generated and released at this deadline.

The network manager is responsible for generating schedules for the field devices deployed in an industrial environment. The schedule for each hyper-period is generated just before the beginning of that hyper-period. In this work, due to the randomness of flow deadlines, we define the hyper-period as the least common multiple of the nearest integer numbers of time slots to the average deadlines of all the flows.

We derive an algorithm for finding the optimal scheduling policy to be administrated by the network manager. This algorithm is obtained by solving the UMDP formulated in Section III.B to obtain a schedule for the next hyper-period. We also introduce a sub-optimal scheduling strategy to reduce the computational complexity of the scheduling process. The sub-optimal algorithm exploits the statistical characteristics of the flows to obtain near-optimal performance.

III. Optimal Scheduling Strategy

In this section, we start by defining the system parameters such as the states, the transition probabilities, and the cost function for the optimization problem to be solved. Then, the UMDP is formulated employing these parameters to introduce the framework for obtaining the optimal scheduling strategy.

A. System Definition

The state of F_m at the beginning of a given time slot is denoted by $S_m = (t_m, h_m)$, where t_m is the number of remaining time slots before the deadline and h_m is the number of remaining hops on the route before reaching the destination. The whole system state which is denoted by $S = (S_1, S_2, ..., S_M)$. In general, if a packet from F_m is scheduled for transmission in a given time slot, S_m will transition to $(t_m - 1, h_m - 1)$ or $(t_m - 1, h_m)$ depending on whether the packet transmission succeeds or fails, respectively. The value of t_m is always greater than zero because a new packet arrival occurs as soon as the previous packet deadline is reached. Hence, t_m changes directly from 1 to the the value d_m of the random deadline D_m for the next packet in F_m. The number of remaining hops h_m takes the value 0 when a packet is successfully received at its destination and moves directly from any value to h_m^* when the packet misses its deadline. The state $S_m = (1, 1)$ is a special state for any m, because one would think the next state if a packet from F_m is scheduled for transmission in a given time slot is (d_m, h_m^*) with d_m representing the value for D_m for the next packet in F_m, regardless of whether the packet transmission succeeds or fails. However, this poses a challenge for computing the total number of missed packets in the network, because such a transition would not distinguish between packet transmission success and failure. We solve this problem by letting the state of the flow transition to the auxiliary state $(d_m, h_m^* + 1)$ if the packet transmission succeeds and to the state (d_m, h_m^*) if it fails. The transitions from $(d_m, h_m^* + 1)$ will be to the states $(d_m - 1, h_m^* - 1)$ and $(d_m - 1, h_m^*)$ depending on whether the packet transmission succeeds or fails, respectively. Various cases are described later during the discussion of state transition probabilities.

The action to be chosen at any time slot is denoted by $A \in \{1, 2, ...M\}$, and it determines which of the flows will be served in this time slot.

The conditional transition probability of the system describes the transition from the state S to the state $\tilde{S}$ given that an action A is chosen. It is denoted by $P(\tilde{S}|S, A)$. The value of this conditional probability is calculated as a function of the conditional probabilities of individual flows as follows

$$P(\tilde{S}|S, A) = \prod_{m=1}^{M} P(\tilde{S}_m|S_m, A), \tag{1}$$

where $P(\tilde{S}_m|S_m, A)$ is the conditional transition probability of S_m given that the action A is chosen. In the following, we provide expressions for these conditional transition probabilities for all the flows given that action $A = n$ is chosen. First, we express the conditional transition probability of S_n given that F_n is scheduled to be served in the current time slot.

$$P\left(\tilde{S}_n|S_n, A = n\right) = \begin{cases} \rho, & (C_1) \text{ or } (C_2), \\ \rho f_n(d_n), & (C_3), \\ 1 - \rho, & (C_4), (C_5), \text{ or } (C_6), \\ (1 - \rho) f_n(d_n), & (C_7), \\ 1, & (C_8) \text{ or } (C_9), \\ f_n(d_n), & (C_{10}) \text{ or } (C_{11}), \end{cases} \tag{2}$$

where, the conditions are defined as follows:

$$(C_1) : \tilde{S}_n = (t_n - 1, h_n - 1), S_n = (t_n, h_n), \min(h_n^*, t_n) \geq \quad h_n > 0, t_n > 1,$$

$$(C_2) : \tilde{S}_n = (t_n - 1, h_n^* - 1), S_n = (t_n, h_n^* + 1), t_n \geq h_n^*,$$

$$(C_3) : \tilde{S}_n = (d_n, h_n^* + 1), S_n = (1, 1), d_n \in \mathcal{B}_n,$$

$$(C_4) : \tilde{S}_n = (t_n - 1, h_n), S_n = (t_n, h_n), t_n > h_n > 0,$$

$$(C_5) : \tilde{S}_n = (h_n - 1, h_n^*), S_n = (h_n, h_n), h_n > 1,$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Hany, Mohamed; Moayeri, Nader.
"Real-Time Scheduling for Wireless Networks with Random Deadlines."
Paper presented at 13th IEEE International Workshop on Factory Communication Systems, Trondheim, Norway. May 31, 2017 - June 2, 2017.

$(C_6) : \tilde{S}_n = (t_n - 1, h_n^*), S_n = (t_n, h_n^* + 1), t_n \geq h_n^*,$

$(C_7) : \tilde{S}_n = (d_n, h_n^*), S_n = (1, 1), d_n \in \mathcal{B}_n,$

$(C_8) : \tilde{S}_n = (t_n - 1, h_n^*), S_n = (t_n, h_n^*), 1 < t_n < h_n^*,$

$(C_9) : \tilde{S}_n = (t_n - 1, 0), S_n = (t_n, 0), 1 < t_n \leq D_n^* - h_n^*,$

$(C_{10}) : \tilde{S}_n = (d_n, h_n^*), S_n = (1, 0), d_n \in \mathcal{B}_n,$

$(C_{11}) : \tilde{S}_n = (d_n, h_n^*), S_n = (1, h_n^*), h_n^* > 1, d_n \in \mathcal{B}_n.$

(C_1) represents the successful transmission of the packet to the following node on the route while the deadline has not been reached yet, and hence the number of remaining hops is decremented by 1. (C_2) represents the successful transmission of a packet at its first transmission from the source when the previous packet was successfully transmitted to the destination at the last time slot before the deadline. (C_3) represents the successful transmission of the packet to its destination at the last time slot before the deadline, and hence the new deadline is a random number following the distribution of $f_n(*)$ and the remaining number of hops is set to h_n^*+1 to indicate successful reception. These three conditions cover all the cases where a packet is successfully transmitted from one network node to another.

Similarly, the conditions for transition probabilities in the cases where the packet transmission fails are as follows. (C_4) represents the case where the transmitted packet has not been successfully received by the following node on the route while the deadline has not been reached yet. (C_5) represents the case where the packet transmission fails resulting in the deadline to be missed because the number of remaining hops is larger than the number of time slots left until the deadline. (C_6) represents the case where the first transmission of the packet from the source fails when the previous packet was successfully transmitted to the destination at the last time slot before the deadline.

Moreover, (C_7) represents the case where the transmission fails and the packet does not make the last hop to the destination and misses its deadline, and hence the new deadline is a random number following the distribution of $f_n(*)$ and the remaining number of hops is set to h_n^* to indicate failed transmission at the previous time slot for this flow.

Next we discuss the cases where no transmission is made when the flow is chosen because either the packet has already missed its deadline (C_8) or been received at the destination before the deadline (C_9).

Furthermore, the conditions (C_{10}) and (C_{11}) represent the cases in which no transmission occurs at the last time slot before the deadline and hence the remaining number of time slots resets to the new random deadline following the distribution of $f_n(*)$. While in the (C_{10}) case the packet has been received at the destination before the deadline, in the (C_{11}) case it has not. Fig. 1 shows the state transitions and the respective probabilities for S_n when the action $A = n$ is chosen.

In general, if a packet from F_n is scheduled for transmission in a given time slot, $S_m = (t_m, h_m)$ for $m \neq n$, will transition to $(t_m - 1, h_m)$ with probability 1. We now provide the complete set of conditional transition probabilities for S_m when F_n for $n \neq m$ is scheduled to be served in the current time slot.

$$P\left(\tilde{S}_m | S_m, A = n\right)|_{\{m \neq n\}} = \begin{cases} 1, & (C_{12}), (C_{13}), (C_{14) \text{ or } (C_{15}), \\ f_m(d_m), & (C_{16}), \end{cases} \quad (3)$$

where, the conditions are defined as follows:

$(C_{12}) : \tilde{S}_m = (t_m - 1, h_m), S_m = (t_m, h_m), t_m > h_m > 0,$

$(C_{13}) : \tilde{S}_m = (h_m - 1, h_m^*), S_m = (h_m, h_m), h_m > 1,$

$(C_{14}) : \tilde{S}_m = (t_m - 1, 0), S_m = (t_m, 0), t_m > 1,$

$(C_{15}) : \tilde{S}_m = (t_m - 1, h_m^*), S_m = (t_m, h_m^*), 1 < t_m < h_m^*,$

$(C_{16}) : \tilde{S}_m = (d_m, h_m^*), S_m = (1, h_m), d_m \in \mathcal{B}_m,$

where (C_{12}), (C_{13}), (C_{14}), and (C_{15}) represent the case in which $t_m > 1$. (C_{12}) represents the case in which the packet in F_m has a remaining number of hops $h_m < t_m$, and hence the packet will continue to be in the network after the current time slot. (C_{13}) represents the case where the packet in F_m has a remaining number of hops $h_m = t_m$, and hence the packet will miss its deadline as a result of not being scheduled for transmission in the current time slot. While (C_{14}) represents the case where the packet in F_m has already successfully arrived at the destination before its deadline. (C_{15}) represents the case where the packet has already missed its deadline.

Finally, (C_{16}) represents the cases where the packet in F_m had already been delivered to its destination, the packet had already missed its deadline, or the packet will miss its deadline as a result of not being scheduled for transmission in this time

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

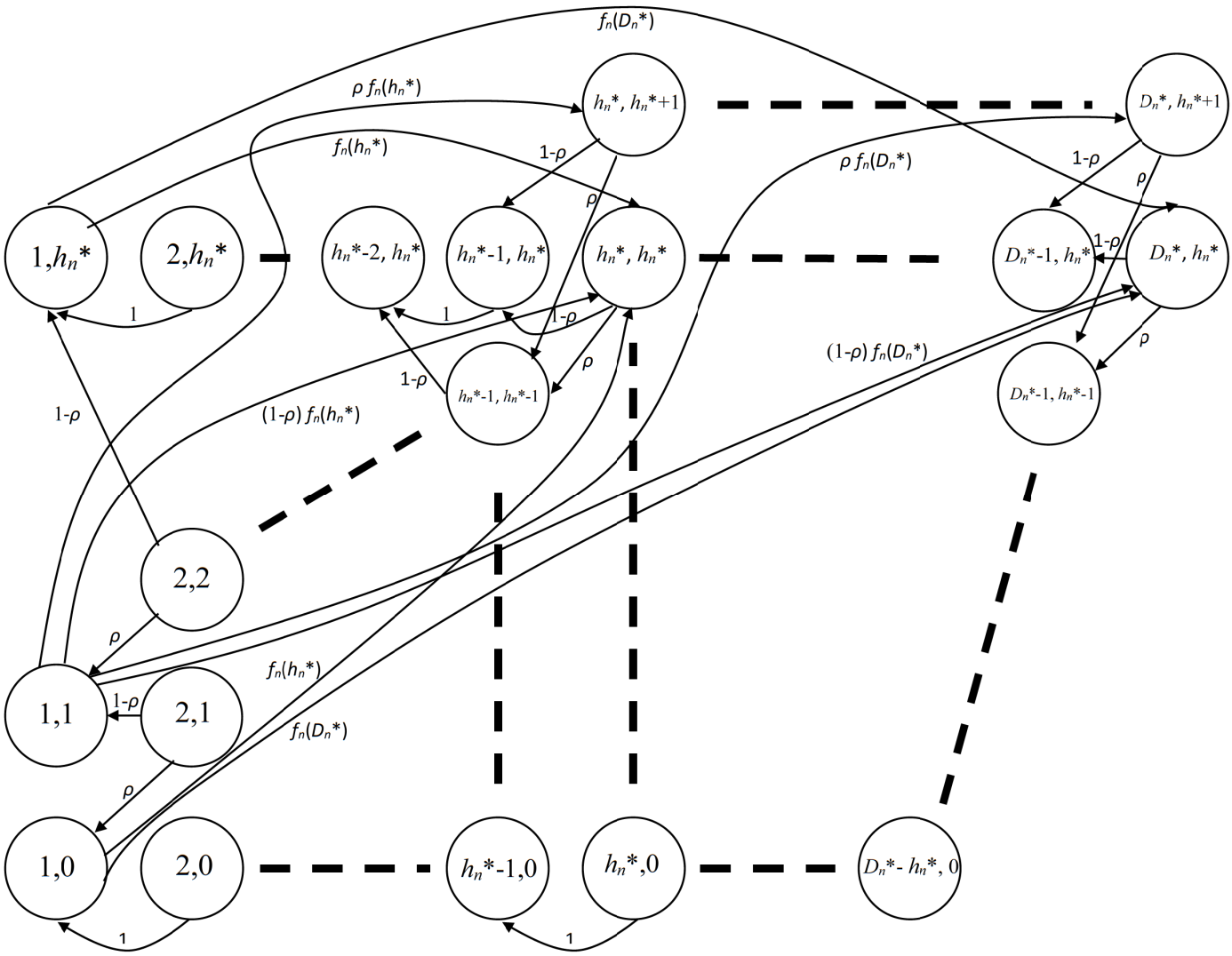


Fig. 1. The Markov chain transitions of the nth flow when $A = n$ is chosen.

slot. In all these cases, a random deadline for a new packet from F_m will be selected following the distribution $f_m(*)$, and the remaining number of hops is set to h_m^*. Fig. 2 shows the state transitions and the respective probabilities for S_m when the action $A = n$, with $n \neq m$, is chosen.

The optimal scheduling strategy is determined by minimizing the total system cost over all possible schedules. The system incurs a cost of one for each packet that misses its deadline for being delivered to its destination. Suppose a packet from F_n, for some $n \in \{1, 2, \ldots, M\}$, is selected for transmission at time slot $t \in \{1, 2, \ldots, T\}$. The system incurs a cost of one in time slot t if F_n is in state (h_n, h_n), for $h_n = 1, 2, \ldots, h_n^*$, and the packet transmission fails. In addition, for each $m \in \{1, 2, \ldots, M\} - \{n\}$, the system incurs a cost of one in time slot t if F_m is in state (h_m, h_m), for $h_m = 1, 2, \ldots, h_m^*$. The transition cost of the system for F_m is denoted by $\gamma\left(\tilde{S}_m | S_m, A = n\right)$, where m and n can take any values in $\{1, 2, \ldots, M\}$. It is given by:

$$\gamma\left(\tilde{S}_m | S_m, A = n\right) = \begin{cases} 1, & (C_{17}) \text{ or } (C_{18}), \\ 0, & \text{Otherwise.} \end{cases} \quad (4)$$

where, the conditions are defined as follows:

$$(C_{17}) : \tilde{S}_m = (h_m - 1, h_m^*), S_m = (h_m, h_m), h_m > 1,$$

$$(C_{18}) : \tilde{S}_m = (d_m, h_m^*), S_m = (1, 1), d_m \in \mathcal{B}_m.$$

The total number of packets that miss their deadlines in time slot $t \in \{1, 2, \ldots, T\}$ is given by:

$$\gamma\left(\tilde{S} | S, A\right) = \sum_{m=1}^{M} \gamma\left(\tilde{S}_m | S_m, A\right), \quad \forall \tilde{S}, S \in \mathcal{S}, A \in \mathcal{A}. \quad (5)$$

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

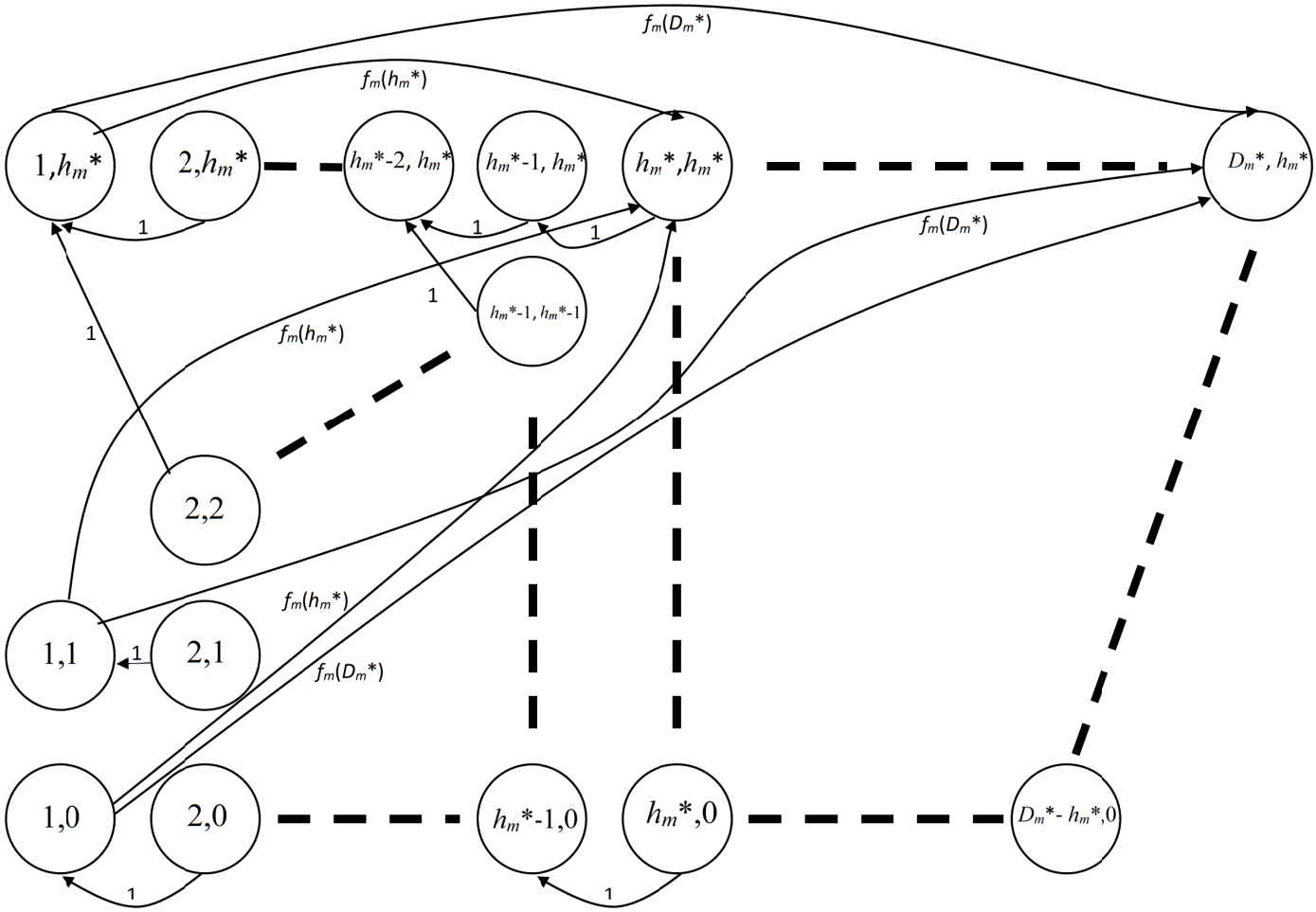


Fig. 2. The Markov chain transitions of the mth flow when $A = n$ is chosen while $n \neq m$.

B. Problem Formulation

In this subsection, we obtain the UMDP formulation for the optimal scheduling problem where the system evolves in a Markovian fashion depending on the chosen scheduling actions at every time slot. Moreover, the schedule is computed at the beginning of the hyper-period and the system state is not observed during the hyper-period. Thus, the problem is considered to be a finite horizon UMDP. The components of the UMDP are defined as follows:

1) The state space, which is denoted by $\mathcal{S}$, contains all the system states S defined in Section III.A.
2) The actions space, which is denoted by $\mathcal{A}$, is defined to be the set of the flows indices $\{1, 2, \ldots, M\}$. The action taken at the beginning of time slot $t \in \{1, 2, \ldots, T\}$ is the flow from which a packet is scheduled for transmission during that time slot.
3) The transition probabilities between system states have been defined in Section III.A.
4) The expected cost is the total number of packets missing their deadlines in time slot t as a result if taking the action $A = n$, as given by Equation (5).

In [12], the author shows that there exists a stationary policy which is optimal for solving POMDP with the average reward/cost criterion under two conditions: 1) the immediate rewards are non-negative; and 2) the Markov chain which represents the system progress is irreducible and ergodic. In our problem, the Markov chain with the network states is irreducible and ergodic. Thus, both conditions are satisfied and the UMDP is a special case of the POMDP.

We define the belief vector P as a probability mass function (pmf) on the set of all system states $\mathcal{S}$, where P_S denotes the probability of a particular system state. The value of this belief vector at the end of a time slot $t \in \{1, 2, \ldots, T\}$ is denoted by $P^{(t)}$. This is the pmf for the set of system states at the end of time slot t. The initial value of this belief vector, denoted by $P^{(0)}$, is assumed to be known through the knowledge of the network state at the beginning of the hyper-period. Moreover, we denote the action taken at a time slot t by $A^{(t)}$, where

a packet from the selected flow is transmitted in time slot t. The complete sequence of actions over the hyper-period is the schedule, which is denoted by $A_{\text{sch}} = \left(A^{(1)}, A^{(2)}, \ldots, A^{(T)}\right)$. In UMDP, the belief update equation for a given time slot depends on the action taken in that time slot only as no information about the state is observed. Hence, the value of the belief vector is updated over time for a given selected action $A^{(t)}$ using the transition probabilities as follows

$$P_{\tilde{S}}^{(t)}|_{A^{(t)}} = P_S^{(t-1)} P(\tilde{S}|S, A^{(t)}), \forall \tilde{S}, S \in \mathcal{S}, A^{(t)} \in \mathcal{A}, \quad (6)$$

where the values of the belief vectors is calculated for all $t \in \{1, 2, \ldots, T\}$ and for all $A^{(t)} \in \mathcal{A}$. We denote the set of all the obtained belief vectors at the time $t \in \{1, 2, \ldots, T\}$ by $\mathcal{P}^{(t)}$.

As the whole schedule A_{sch} is computed and fixed at the beginning of the hyper-period and the network is not observed during the hyper-period, the optimal schedule, denoted by A^*_{sch}, can be evaluated directly by solving over the complete sequence of actions as follows

$$A^*_{\text{sch}} = \underset{\left(A^{(1)}, A^{(2)}, \ldots, A^{(T)}\right) \in \mathcal{A}^T}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{\tilde{S}} \sum_{S} P_S^{(t-1)} \cdot \gamma(\tilde{S}|S, A^{(t)}) P(\tilde{S}|S, A^{(t)}). \quad (7)$$

This is a very high-dimensional optimization problem depending on the hyper-period length. As a result, we will use POMDP-solving sequential techniques in order to efficiently solve the formulated problem.

In order to evaluate the optimal strategy, a backward tracing is performed over the objective function where the initial value at the last time slot of the hyper-period is evaluated as follows

$$V_{A^{(T)}}^{(T)}(P^{(T-1)}) = \sum_{\tilde{S}} \sum_{S} P_S^{(T-1)} \gamma(\tilde{S}|S, A^{(T)}) P(\tilde{S}|S, A^{(T)}), \quad \forall A^{(T)} \in \mathcal{A}, P^{(T-1)} \in \mathcal{P}^{(T-1)}. \quad (8)$$

Then, the objective is evaluated using backward tracing by adding the minimum cost for all future time slots to the immediate cost at the current time slot. The recursive relation for calculating the cost function is expressed as follows

$$V_{A^{(t)}}^{(t)}(P^{(t-1)}) = \sum_{\tilde{S}} \sum_{S} P_S^{(t-1)} \gamma(\tilde{S}|S, A^{(t)}) P(\tilde{S}|S, A^{(t)}) + \min_{A^{(t+1)} \in \mathcal{A}} V_{A^{(t+1)}}^{(t+1)}(P^{(t)}), \quad \forall A^{(t)} \in \mathcal{A}, t \in 1, 2, \ldots, T-1, P^{(t-1)} \in \mathcal{P}^{(t-1)}. \quad (9)$$

The optimal action at time slot t is evaluated using the obtained objective functions and belief vector at the edge of the time slot t as follows

$$A^{*(t)}(P^{(t-1)}) = \arg \min_{A^{(t)} \in \mathcal{A}} V_{A^{(t)}}^{(t)}(P^{(t-1)}), \forall P^{(t-1)} \in \mathcal{P}^{(t-1)}. \quad (10)$$

The above recursive equations can be solved sequentially as follows: i) Solve Equation (6) for all $t \in \{1, 2, \ldots, T\}$ and for all $A^{(t)} \in \mathcal{A}$, ii) evaluate Equation (8), iii) solve the optimization problem (10) at the time slot corresponding to the last evaluated objective function, iv) evaluate Equation (9) for the preceding time slot, and v) go back to item iii) except when $t = 1$. After running this described algorithm, the optimal schedule can be expressed as follows

$$A^*_{\text{sch}} = \Big(A^{*(1)}(P^{(0)}), A^{*(2)}(P^{(1)}|_{A^{*(1)}}), \ldots, A^{*(T)}(P^{(T-1)}|_{A^{*(T-1)}})\Big). \quad (11)$$

In this paper, the formulated UMDP is then solved using one of POMDP solving techniques used by the solver in [13] to obtain the optimal schedule of the system in a computationally efficient way compared to implementing the backward tracing algorithm directly.

IV. A Sub-Optimal Scheduling Policy

In the case of periodic flows with deterministic deadlines, various scheduling strategies have been considered [14]. A common real-time scheduling policy that has been found to be effective for industrial environments is the earliest deadline first (EDF) strategy [15]. It has been shown in [11] that it outperforms fixed priority scheduling, where the priorities of flows are fixed over time and do not depend on the deadlines, while delivering competitive acceptance ratios to the optimal dynamic priority scheduling policies, where the priorities depend on the network status and can change with time, at lower computational cost. Moreover, EDF has been used in [11] for obtaining schedules over a hyper-period in industrial environments.

In this work, we propose an extension to the EDF strategy to work in the case of random deadlines. The proposed strategy results in a sub-optimal schedule for a whole hyper-period of the discussed network. We denote the proposed strategy by earliest average deadline first (EADF). The computational complexity of EADF is much smaller than that of the optimal scheduling scheme. In this strategy, we take into account the averages of the flow deadlines to choose the transmitting flow at a time slot. At any time slot, the flow with the earliest expected deadline is chosen for transmission. Algorithm 1 illustrates the proposed technique as follows

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Algorithm 1 EADF Scheduling Strategy

Initialization: $T, M, \mu_m \forall m$;
Initialization: $S^{(0)}$;
for $t = 1 : T-1$ **do**
  Evaluate: $A^{*(t)} = \arg\min_m t_m^{(t-1)}$;
  for $m = 1 : M$ **do**
    if $t_m^{(t-1)} > 1$ **then**
      Update: $t_m^{(t)} = t_m^{(t-1)} - 1$;
    else
      Update: $t_m^{(t)} = \text{round}(\mu_m)$;
    end if
  end for
end for
Output: $A^*_{\text{sch}} = \left(A^{*(1)}, A^{*(2)}, \ldots, A^{*(T)}\right)$;

V. Numerical Results

In the following, we assess the performance of the proposed scheduling algorithms when packets with random deadlines are considered. The minimized objective function is the total number of packets that have missed their deadlines. We will compare the performance of the optimal strategy, the EADF strategy, and the basic round robin benchmark in which flows are scheduled over time in equal portions and in circular order without prioritizing any of the flows [16]. We will refer to these three strategies, respectively, as 'Optimal', 'EADF', and 'Round Robin'. In these simulations, we have used the POMDP solver in [13] to obtain the optimal scheduling strategy. In all the following results, the deadlines are following a uniform distribution over the range $[\max(\mu_m - 1, h_m^*), \mu_m + 1]$.

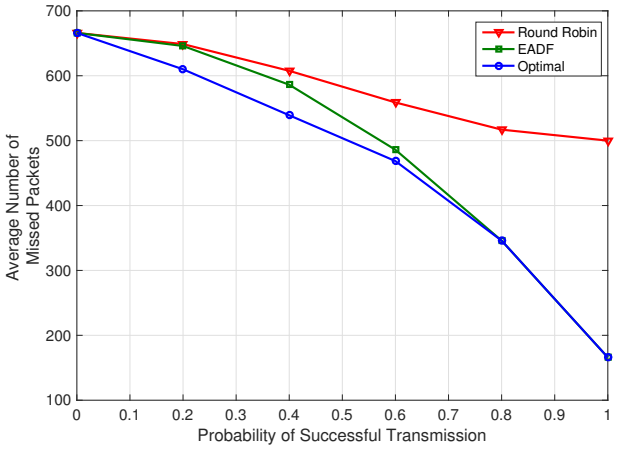


Fig. 3. The total number of missed packets against the channel quality for various scheduling strategies.

In Fig. 3, we consider the case of two flows with $h_1^* = h_2^* = 2$, $\mu_1 = 2$, $\mu_2 = 6$, and $T = 1000$. We have chosen a long hyper-period to capture the effect of randomness on the performance. We show the performance as a function of success probability in transmitting a packet. The improvement in the performance of the optimal policy over the round robin policy is higher in the case of better channels. EADF's performance is identical to the optimal policy for success probability larger than 0.8 and only slightly worse for low to moderate values of success probability. The EADF strategy actions depend on the number of remaining time slots to the deadline and not the remaining number of hops. As a result in the case of poor channels, the EADF does not prioritize flows with lower remaining number of hops to improve packet delivery for the flows.

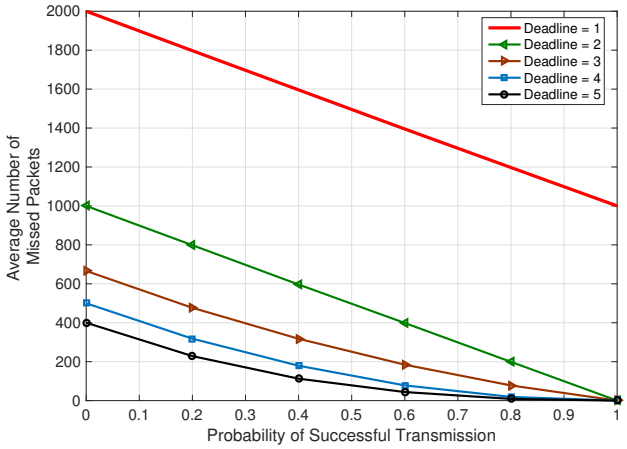


Fig. 4. The total number of missed packets against the channel quality for various average deadline values.

In Fig. 4, we consider the case of 2 symmetric flows with a single hop between the respective sources and destinations. We show the system performance for different values of the same average deadline for both flows. As the average deadline increases, the improvement in the performance decreases. Also, this improvement is larger for poorer channel conditions. This shows the importance of having good scheduling schemes for poor conditions due to the necessity to optimize system parameters for improved performance.

In Fig. 5, we consider two flows with $h_1^* = 1$ and $h_2^* = 3$ and a common average deadline μ. We compare the performance of the optimal, EADF and round robin policies for two values of μ, namely 3 and 5. At $\mu = 3$, the performance of the optimal policy almost coincides with that of the round robin strategy at $\rho = 1$ because, on average at every 3 time slots, a packet arrives at each of the flows, $h_1^* = 1$, and $h_2^* = 3$. As a result, only one packet from these two flows can be delivered before its deadline in almost all of the cases and hence both algorithms have very similar total number of packets missing

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

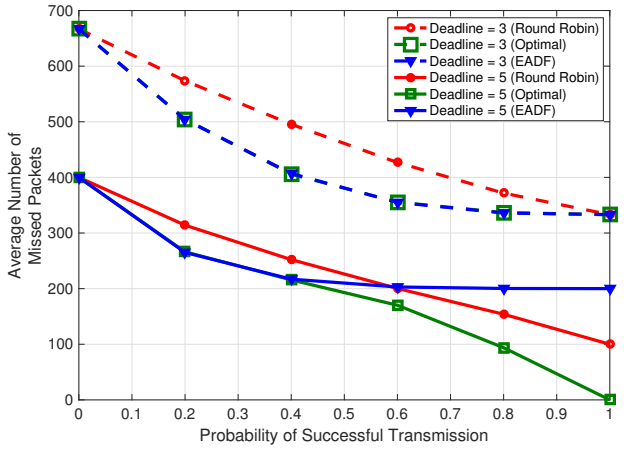


Fig. 5. The total number of missed packets against the channel quality for various average deadline values and verious scheduling strategies.

their deadlines. Moreover, the EADF performance coincides withe optimal for the whole range of ρ. At $\mu = 5$, the optimal strategy achieves better performance because both flows can be scheduled within five time slots successfully, and hence an efficient scheduling scheme improves the performance significantly. Moreover, the EADF has a poor performance at the high values of ρ because it does not consider channel quality during the schedule calculation and hence it does not benefit from the good channel quality while it has optimal performance on the poor channel quality cases.

VI. Conclusions

In this work, we have modeled the scheduling problem for a system with M flows with random deadlines as a UMDP. The investigation of the case of flows with random deadlines has been motivated by scenarios in industrial environments where data flows carry sensing and control decision data. The solution of the UMDP is the optimal scheduling strategy which determines the sequence of chosen flows for transmission over the hyper-period. We also obtained a sub-optimal EADF strategy to lower the complexity of the scheduling process. Numerical results suggest the use of the optimal policy in situations where the channel quality is good and the average deadlines for the flows are large compared to the lengths of the routes of various flows. Also, it is beneficial to use the optimal scheduling policy for tighter deadlines when the channel quality is poor.

References

[1] M. Nixon, “A comparison of WirelessHART and ISA100.11a,” Technical Report MSU-CSE-06-2; Emerson Process Management: Round Rock, TX, USA, 2012.

[2] W. Liang, X. Zhang, Y. Xiao, F. Wang, P. Zeng, and H. Yu, “Survey and experiments of WIA-PA specification of industrial wireless network,” Wirel. Commun. Mob. Comput. 2011, 11, 1197-1212.

[3] S. Kumar, and S. Chauhan, “A survey on scheduling algorithms for wireless sensor networks.” Int. J. Comput. Appl. 2011, 20, 713.

[4] M. Chitnis, P. Pagano, G. Lipari, and Y. Liang, “A survey on bandwidth resource allocation and scheduling in wireless sensor networks,” In Proceedings of the International Conference on Network-Based Information Systems, Gwangju, Korea, 4 September 2009; pp. 121128.

[5] S. Rao, S. Keshri, D. Gangwar, P. Sundar, and V. Geetha, “A survey and comparison of GTS allocation and scheduling algorithms in IEEE 802.15.4 wireless sensor networks,” In Proceedings of the IEEE Conference on Information Communication Technologies, JeJu Island, Korea, 11 April 2013; pp. 98103.

[6] L. Kruk, J. Lehoczky, S. Shreve, and S.-N. Yeung, ”Earliest- deadline-first service in heavy-traffic acyclic networks,” The Annals of Applied Probability, vol. 14, no. 3, pp. 1306-1352, 2004.

[7] Z. Mao, C. E. Koksal and N. B. Shroff, ”Optimal Online Scheduling With Arbitrary Hard Deadlines in Multihop Communication Networks,” in IEEE/ACM Transactions on Networking, vol. 24, no. 1, pp. 177-189, Feb. 2016.

[8] H. F. Zhu, J. P. Lehoczky, J. P. Hansen and Ragunathan Rajkumar, ”Diff-EDF: a simple mechanism for differentiated EDF service,” 11th IEEE Real Time and Embedded Technology and Applications Symposium, 2005, pp. 268-277.

[9] H. Li, P. Shenoy and K. Ramamritham, ”Scheduling messages with deadlines in multi-hop real-time sensor networks,” 11th IEEE Real Time and Embedded Technology and Applications Symposium, 2005, pp. 415-425.

[10] R. Fox, and M. Tennenholtz, “A reinforcement learning algorithm with polynomial interaction complexity for only-costly-observable MDPs,” Proceedings of AAAI-07, 2007.

[11] C. Wu, M. Sha, D. Gunatilaka, A. Saifullah, C. Lu, and Y. Chen, “Analysis of EDF scheduling for wireless sensor-actuator networks,” In IWQoS, 2014.

[12] R. Cavazos-Cadena, “Weak conditions for the existence of optimal stationary policies in average Markov decision chains with unbounded costs,” Kybernetika, vol. 25, no. 3, pp. 145156, 1989.

[13] POMDP Solver: Available Online: http://www.pomdp.org/

[14] M. Nobre, I. Silva, and L. Guedes, “Routing and scheduling algorithms for WirelessHART networks: A survey,” Sensors, vol. 15, no. 5, p. 9703-2015.

[15] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, “Real-time scheduling for WirelessHART networks, in RTSS10.

[16] M. Shreedhar, and G. Varghese, “Efficient fair queuing using deficit round robin,” In ACM SIGCOMM 95 (1995).

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

An Analysis of Vulnerability Trends, 2008 - 2016

D. Richard Kuhn[1], M S Raunak[2], Raghu Kacker[1]
kuhn@nist.gov, raghu.kacker@nist.gov raunak@loyola.edu
[1]National Institute of Standards and Technology [2]Loyola University of Maryland

Computer security has been a subject of serious study for at least 40 years, and a steady stream of innovations has improved our ability to protect networks and applications. But attackers have adapted and changed methods over the years as well. Where do we stand today in the battle between attackers and defenders? Are attackers gaining ground, as it often seems when reading press accounts of the latest data exposure? This analysis seeks to answer these questions using data from the US National Vulnerability Database (NVD) [1], and to identify classes of vulnerabilities where improvements will be most cost effective.

Data. The NVD is the US government's repository of information system security vulnerabilities. It is operated by the US National Institute of Standards and Technology, and is sponsored by the Department of Homeland Security's National Cyber Security Division. The NVD relies on publicly reported vulnerabilities from the Common Vulnerabilities and Exposures (CVE) dictionary. As of Spring 2017, there are more than 83000 vulnerabilities enumerated in the database. The NVD adopted Version 2.0 of the Common Vulnerability Scoring System (CVSS) in June 2007 to score the severity of each reported vulnerability, prior to the period in which this analysis begins. To ensure maximum consistency of data scoring and definition, we have used only reports from the period 2008 to 2016.

Vulnerability Severity. One area in which some progress is apparent is in the severity of vulnerabilities that are being discovered. For the NVD, severity is rated using the CVSS, which combines scores for impact and exploitability. As can be seen in Table I and Fig. 1, the proportion of high severity vulnerabilities is trending downward, declining about 15 percentage points since 2008. About two-thirds of this fraction has shifted to Medium severity vulnerabilities, which increased from about 46% to 55% of the total, while Low severity numbers increased from 3% to nearly 10% of the total.

Vulnerability Types. Table II shows the primary vulnerability categories used in the NVD. Each reported CVE is assigned to one or more categories called the Common Weakness Enumeration (CWE). Some of these primary CWE categories may include a number of subsidiary weaknesses. For example, CWE-119, Buffer errors, includes 14 subsidiary CWEs, such as out of bounds read (CWE-125), and untrusted pointer dereference (CWE-822). NVD entries in the 2008 to 2016 period were categorized as one of these types, with the exception of some which could not be determined because of insufficient information.

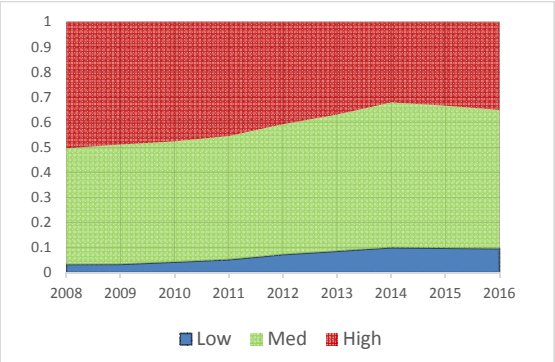


Fig. 1. Vulnerability Severity Trends, 2008-2016

TABLE I. VULNERABILITY SEVERITY, 2008-2016

	Low	Med	High
2008	0.033	0.463	0.504
2009	0.034	0.477	0.489
2010	0.043	0.481	0.477
2011	0.052	0.493	0.455
2012	0.073	0.519	0.408
2013	0.086	0.544	0.369
2014	0.100	0.579	0.322
2015	0.098	0.568	0.334
2016	0.096	0.553	0.350

TABLE II. NVD VULNERABILITY CATEGORIES

CWE-ID	Description	Type	Trend
CWE-16	Configuration	C	↓
CWE-20	Input Validation	I	↑
CWE-22	Path Traversal	I	↓
CWE-59	Link Following	I	≈
CWE-78	OS Command Injections	I	↑
CWE-79	Cross-Site Scripting (XSS)	I	≈
CWE-89	SQL Injection	I	↓
CWE-94	Code Injection	I	↓
CWE-119	Buffer Errors	I	↑
CWE-134	Format String Vulnerability	I	≈
CWE-189	Numeric Errors	I	↓
CWE-200	Information Leak / Disclosure	C	↑
CWE-255	Credentials Management	D	↑
CWE-264	Permissions, Privileges, Access	D	↑
CWE-287	Authentication Issues	D	≈
CWE-310	Cryptographic Issues	D	↑
CWE-352	Cross-Site Request Forgery	I	≈
CWE-362	Race Conditions	I	↑
CWE-399	Resource Management Errors	I	↓

We grouped the NVD CWE classes into primary types of Configuration, Design, and Implementation errors, designated in Table II as C, D, and I respectively. Table II also indicates whether the different vulnerability types are increasing (↑), decreasing (↓), or approximately unchanged (≈). In determining the type of each CWE class, we considered the common errors in each type. Configuration vulnerabilities result when a system is not set up correctly with respect to security goals. A simple example would be failure to enable password checking. Information leak is a broader type, but in most cases,

Kacker, Raghu; Kuhn, David; Raunak, Mohammad.
"An Analysis of Vulnerability Trends, 2008 - 2016."
Paper presented at IEEE International Conference on Software Quality Reliability and Security,
Prague, Czech Republic. July 25, 2017 - July 29, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

available security controls have been neglected or set up improperly, so this is designated as a Configuration error. Design-related vulnerabilities are those that originate in the planning and design of the system, such as selecting an outdated or weak cryptographic algorithm. The third source of vulnerabilities is typically simpler, but may have dramatic results. One of the most common implementation vulnerabilities is the simple buffer overflow. Failure to check that input size is within maximum buffer size is a simple error that should almost never occur, but continues to be a widespread problem (Table III). Some categories are less obvious. For instance cross-site scripting can have several forms, but in each case results from missing or inadequate input validation, so this is also included in implementation errors. Most of the other implementation-related vulnerabilities in Table II also result from failure to properly validate input.

What is most striking about the distribution of Configuration, Design, and Implementation errors captured in Fig. 2 is that implementation or coding errors account for roughly two thirds of the total. We consider the proportion of implementation vulnerabilities, rather than absolute numbers, because the number of vulnerabilities is partially a function of the number of applications released, which has increased over time. The proportion of implementation vulnerabilities for 2008 to 2016 is close to the 64% reported for 1998 to 2003 in an analysis of an early version of NVD [2]. This suggests that little progress has been made in reducing these vulnerabilities that result from simple mistakes which should be easy to prevent.

But this also means there is potential for significant reductions in vulnerabilities. Clearly better testing could prevent most such simple errors from making it into a released product, and practices such as code reviews and static analysis checks can be especially cost-effective for simple errors. Static analysis has been shown to detect about 20% of CVE-defined errors [3], and formal code inspection may prevent an average of about 65% of errors from reaching released products [4]. Thus vulnerabilities could be reduced with broader use of such practices.

To see the potential for improving cybersecurity through basic development practice, consider the absolute numbers of vulnerabilities shown in Table III (cryptographic issues adjusted for a spuriously large number in 2014 due to multiple entries resulting from failure to check X.509 certificates in Android apps). Implementation errors are highlighted in bold type; they represent a total of 27 242 of the 37 325 categorized vulnerabilities, or 72.9% for the 2008-2016 period. Note in particular that two of the presumably simplest errors to prevent, basic input validation and buffer errors, account for more than a third of the implementation flaws.

While the basic recommendations in this paper, greater use of static analysis tools and code review, have been made many times in the past [2], we note that progress has been made in static analysis, notably in the reduction of false positives and improved detection [3] [5], and code review is consistently shown to be highly cost effective [6]. This analysis will be extended to review trends within the different vulnerability types and subsidiary weaknesses, with a goal of identifying practices that may have the strongest impact on reducing vulnerabilities.

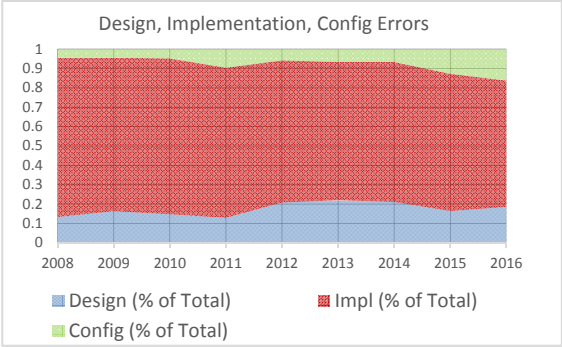


Fig. 2. Vulnerability Class Trends, 2008-2016

TABLE III. VULNERABILITY COUNTS FOR 2008-2016

Format String Vulnerability	**110**
Configuration	195
OS Command Injections	**208**
Race Conditions	**377**
Link Following	**389**
Credentials Management	589
Cryptographic Issues	779
Authentication Issues	920
Cross-Site Request Forgery (CSRF)	**1161**
Numeric Errors	**1199**
Code Injection	**1545**
Path Traversal	**1686**
Information Leak / Disclosure	2939
Input Validation	**3763**
SQL Injection	**3828**
Permissions, Privileges, and Access	4661
Cross-Site Scripting (XSS)	**6220**
Buffer Errors	**6756**
Total	37325

Products may be identified in this document, but such identification does not imply recommendation by the US National Institute of Standards and Technology or the US Government, nor that the products identified are necessarily the best available for the purpose.

[1] National Vulnerability Database, http://nvd.nist.gov 2017
[2] Heffley, Jon, and Pascal Meunier. "Can source code auditing software identify common vulnerabilities and be used to evaluate software security?" *System Sciences, 37th Annual Hawaii Intl Conf*, IEEE, 2004.
[3] Okun, Vadim, Aurelien Delaitre, and Paul E. Black. "Report on the static analysis tool exposition (SATE) IV" *NIST Special Publication* 500 (2013): 297.
[4] Jones, C, "Measuring Defect Potentials and Defect Removal Efficiency", *Crosstalk, Journal of Defense Software Engineering* (June 2008).
[5] Medeiros, I., Neves, N. and Correia, M., 2016. Detecting and removing web application vulnerabilities with static analysis and data mining. *IEEE Trans. Reliability*, *65*(1), pp.54-69.
[6] Balachandran, V., 2013, May. Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation. In *Software Engineering (ICSE), 2013 35th International Conference on* (pp. 931-940). IEEE.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Combinatorial Testing of Full Text Search in Web Applications

M S Raunak[1] raunak@loyola.edu
D. Richard Kuhn[2] kuhn@nist.gov
Raghu Kacker[2] raghu.kacker@nist.gov
[1]Loyola University of Maryland
[2]National Institute of Standards and Technology

Abstract: **Database driven web applications are some of the most widely developed systems today. In this paper, we demonstrate use of combinatorial testing for testing database supported web applications, especially where full-text search is provided or many combinations of search options are utilized. We develop test-case selection techniques, where test strings are synthesized using characters or string fragments that may lead to system failure. We have applied our approach to the National Vulnerability Database (NVD) application and have discovered a number of "corner-cases" that had not been identified previously. We also present simple heuristics for isolating the fault causing factors that can lead to such system failures. The test method and input model described in this paper have immediate application to other systems that provide complex full text search.**

I. INTRODUCTION

Web-based applications, especially ones driven by a back-end database, continue to be some of the most common custom software being developed in today's world. Many applications store their data in an SQL database or in a no-SQL data store at the back-end servers and query and update them to provide information searched by the users and other transaction-oriented service. The architecture of these applications is usually layered, utilizing many different, often loosely connected components. Testing these applications effectively remains a challenge for the software engineering community. High profile web application failures have resulted in cases where testing was insufficient [2][3]. A wide variety of general-purpose strategies and techniques for systematically testing applications are well established and widely practiced [1].

In this paper we focus on a highly specific test procedure for full-text search in the National Vulnerability Database (NVD), a large, heavily used public internet database. Text search is one of the fundamental components of web applications used in every industry. Thus, strong testing of this component is essential for high assurance, but it is often handled as simply one aspect of overall system testing. The test procedure documented in this paper seeks to provide stronger testing for this fundamental component.

After implementing a new feature in the NVD, developers discovered that certain special characters resulted in "Server Error" responses. However it was not clear which specific combinations of special characters triggered this response, or how many such problematic cases existed. Some of these were corrected, but it was decided to apply combinatorial methods to attempt to identify all inputs that caused the server error response.

Decision makers in industry and government rely on testing to determine readiness of systems for deployment, so defensible measures of test completeness are essential. Testing any reasonable application exhaustively is nearly always impractical. The two crucial questions testing researchers aim to study are how to select the test cases, and how many to select; i.e., when to stop testing. These questions become especially challenging for integration and system testing of any large application. The goal is to select test cases at the integration and system testing level such that the fault finding probability is maximized. One can try to achieve this goal by systematically covering a large section of the input space including many corner cases or unexpected values that may potentially cause failures.

Combinatorial Testing [4] has been shown to be an effective approach. In this paper, we demonstrate use of combinatorial approaches to develop test cases that systematically test important components of database-backed web applications. Our case study reveals that such systematic exploration of input space using covering arrays can be a very useful way of identifying failure scenarios that are otherwise not discovered.

Web applications are typically developed in a multi-tiered fashion. Fig. 1 shows a high level generic architecture of most of these applications. The outermost layer is the presentation layer, which provides the interface for client interaction. Application developers use HTML, CSS, and client side scripting languages to make the user interface intuitive and useful. The Middle layer is often served by an Application Server framework such as JBoss, WebSphere or Netweaver. These application servers are often built around common web servers such as Apache and IIS. Programmers develop their business logic and run their programs on these Application Server environments. At the innermost layer sits a DBMS such as mySQL, MS SQL, Oracle, DB2 etc. The application servers provide support for interacting with the DBMS through standard protocols like Java Database Connectivity (JDBC), Common Object Request Broker Architecture (CORBA) etc.

While developing a web application, there are programming modules that are developed at each level. Programmers often write stored procedures, which are

Kacker, Raghu; Kuhn, David; Raunak, M S.
"Combinatorial Testing of Full Text Search in Web Applications."
Paper presented at IEEE International Conference on Software Quality Reliability and Security,
Prague, Czech Republic. July 25, 2017 - July 29, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

groups of SQL statements stored and executed in the RDBMS, i.e., at the innermost layer, to reduce server-client

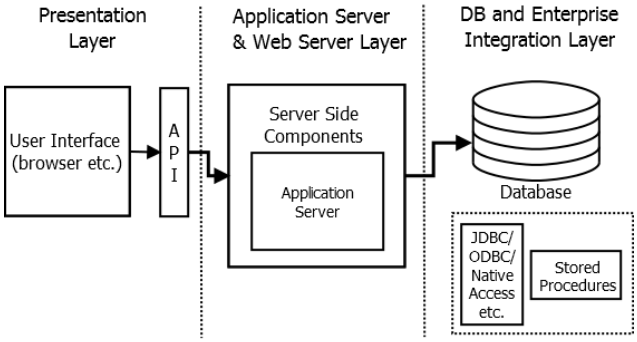


Fig. 1. A Typical Multi-Layered Web Application

network traffic, and to make the system more secure by hiding many of database-level details. The middle layer, where most of the business logic is implemented, often includes integration of other already developed modules or service calls to independent web services. Similarly, for receiving and reacting to user interaction and presenting the output of the application to the client, program modules are developed and deployed to interact with the middle layer directly. All these component interactions, both at the same layer and across layers, require systematic and thorough testing. Programmers and managers are usually good at making sure that unit tests are developed and run regularly while different programming units are being developed. However, testing unit interactions during integration, and testing the overall system systematically once it has been developed, often receive relatively little attention. Anecdotal evidence suggests that the primary reason for this relative lack of integration and system testing is often deadline pressure and not allowing enough time for testing during project estimation and planning phase. A second factor that makes this scenario more challenging is the fact that a systematic process of testing these types of application is not yet well established. A third challenge comes from the often dynamic nature of the underlying data, which makes it more difficult to develop test oracles to support test cases that can be automated.

In this study, we explore a systematic way of testing certain system level testing of database backed web applications and report the effectiveness of our approach.

II. Related Work

Because of their practical importance, database applications have been the subject of a variety of investigations of combinatorial test methods. In particular, combinatorial testing is especially appropriate because database systems must parse and interpret complex queries structured as regular expressions or predicates [15]. In addition to the complexity of the inputs, database applications are also characterized by the need to test both the database functions and the database system interaction with the application that accesses the database [20]. The test problem is further complicated by dependence on the initial database state, which may influence structural coverage metrics, i.e., the degree to which the code can be exercised [8] The complexity of database queries has led to the need for specialized coverage metrics that include the evaluation of conditions in search predicates [7].

Some research has shown that the distribution of *t*-way faults in MySQL database applications is similar to many other application domains, following the interaction rule that most faults are caused by a single factor or two factors interacting, with progressively fewer by 3-way or higher strength interactions [19]. The empirical data showed that a significant proportion of SQL faults involved 3-way or higher strength interactions, suggesting the need for combinatorial methods. Pairwise testing was shown to be effective [15] for discovering many bugs not detected by conventional test methods. It was shown that CT detected a wide range of previously undiscovered faults in a web based database using 2-way through 4-way testing [16]. These methods were also shown to be effective for testing security of database applications [17][18].

Also relevant to our work are investigations comparing the effectiveness of covering arrays with random test generation. This question has been studied in a variety of contexts. In many cases the comparison between combinatorial and random methods has considered only pairwise test arrays, with an equal number of randomly generated tests. Schroeder et al.[22] compared randomly generated tests with *t*-way arrays for $t = 2, 3, 4$. Covering arrays were generated using a tool called TVG, and applications tested had input model configurations of $2^{16}5^{1}8^{1}$ and $2^{7}3^{10}4^{2}$. Because the covering arrays were large, random test sets of the same size covered 95% to 99.99% of the *t*-way combinations, and there was no significant difference between *t*-way testing and random testing. This example illustrates the point that coverage of combinations is a key consideration, whether this is achieved by covering arrays or other test generation methods. Another study [23] found than manually constructed tests could be more effective than 2-way test arrays, but at higher strengths there was no difference, and results from randomly generated tests were not consistent. A number of studies have consistently found covering arrays to be more effective than random tests, including [24][25][26], which investigated the testing of logical expressions.

Others showing significantly better results for *t*-way testing include [27][28][29]. Two key considerations must be evaluated in comparing the two approaches to testing: combinatorial coverage of test sets, and input model design. It is easy to show that a large enough randomly generated

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

test set will cover a high proportion of *t*-way combinations, so the comparison between covering arrays and random test generation is largely a question of efficiency, at least at lower interaction strengths. For *t*-way testing of 4-way and above, a random test set covering the same proportion of combinations may be prohibitively large. The importance of the input model can be seen in research that demonstrates significant differences in structural coverage and fault-detection effectiveness as the input model is changed. Examples include [30], where branch coverage was increased from roughly 70% to 100% only through input model changes, and [31], which demonstrated improved fault detection results for both covering arrays and random tests depending on the input model used.

III. Approach

Our approach is to search for failure scenarios through systematic coverage of the input space and user interactions at the system testing level. For database backed web applications, especially the ones that primarily render some subset of data for information purposes, one of the major components is some sort of query functionality. Users are provided an interface to query the underlying information in many different ways. Consider the case of searching the catalogs of any library or the flight search in a travel application on the web. An important testing aspect in these scenarios is to verify that the search functionality behaves as expected under all circumstances.

In addition to testing for expected functional behavior, any application, especially the ones available over the web, also needs to be tested for potential failure scenarios against unexpected input. If there are unintentionally mistyped or maliciously created inputs that can cause system failure or unexpected behavior, then it also becomes a security issue, which demands attention. Developers and testers often test only the most common expected interaction from users, which is also commonly known as `Happy Path Testing'. This testing practice leaves out the necessary aspect of searching for system failures under unexpected user inputs or interactions.

We use a combinatorial approach and come up with covering array of a wide range of input combinations. We utilize these covering arrays to create test cases. For this particular study, we focus on primarily two types of test-scenarios: a) user inputted strings that may cause failures, and b) user selected options in a web form.

IV. Case Study

To apply our approach of developing effective test cases using combinatorial coverage for systematically testing database-backed web application, we selected the National Vulnerability Database or NVD [12] project. NVD is a project under the Computer Security Division of the National Institute of Standards and Technology (NIST). It maintains a repository of publicly known hardware and software vulnerabilities in a standardized fashion. Every vulnerability is uniquely identified by a CVE-ID (Common Vulnerability and Exposure Id), which is primarily assigned by the MITRE corporation and, to a limited degree, by some other CVE Numbering Authorities (CNAs) [13]. Once a reported vulnerability has been assigned a CVE-ID, it finds its way to the NVD group at NIST. Here the submitted CVE is thoroughly analyzed for standardization and is placed under one or more CWE (Common Weakness Enumeration) categories. Additionally, NVD analysts checks all the references, standardizes different aspects of the vulnerability description, and assign a severity score to the vulnerability following Common Vulnerability Scoring System or CVSS [14]. Once a new vulnerability has been standardized, categorized, and reference-checked, they are made available for public use through the NVD web site. This NVD data provides support for many valuable services such as enabling automation of vulnerability management, security measurement, and compliance (e.g. FISMA).

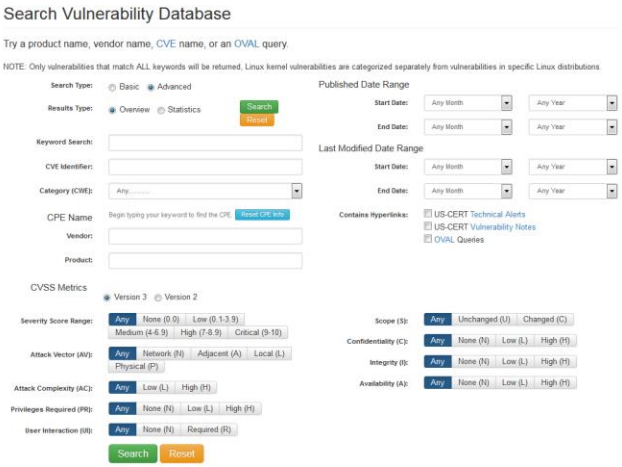


Fig. 2. Advanced Search Option of NVD Data

The NVD website provides a user interface for looking up information about all the CVEs and their corresponding information stored in its data set. In the base search form, there is only an option to perform keyword search. The NVD application looks for whatever the user has inputted in its database and shows results for entries with the search string in them. NVD also provides an advanced search option. In this web form, it allows users to search for any keyword, CWE category, CVE-Id as well as a large number of different options such as date-range (month and year) and CVSS scores. Fig. 2 shows an image of the web form for the advanced search page. In both the basic and advanced search options, there is an option for keyword search in this search function. The user can type in search phrases like “buffer overflow”, “X 509”, “Android”, or “2.3”, to look up vulnerabilities that match these keywords. In the advanced search page, a user can choose a CWE category from a

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

drop-down list and search for all the CVEs that have been categorized under that CWE. There are also a number of fine grained search options related to the different fields of a CVSS 2.0 or 3.0. When the user chooses these additional search criteria, the search functionality queries its database and returns the number of CVEs that met the criteria. If there are hits, the CVEs are listed as shown in Fig. 3.

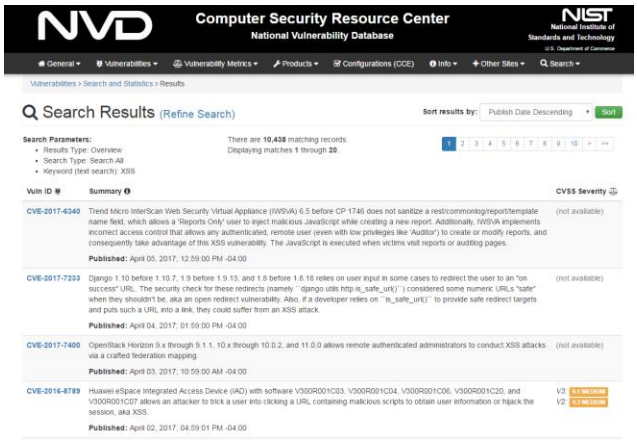


Fig. 3. Search Result Page of NVD

A. Combinatorial Input Model for Search Strings

One of our objectives for this study has been to systematically discover if there are search strings that may result in unexpected behavior from the NVD system. For any web application, coming up with an effective set of search strings to test the search functionality (e.g., keywords search) of the system is one of the common challenges for any test designer.

Instead of focusing only on likely keywords that users may use in a search, we approached the problem with a goal of creating keywords that are a combination of expected inputs such as simple strings and potentially unexpected symbols. Our hypothesis is that how the system responds to such rarely used search strings may not have thoroughly been tested for many web applications.

Parameter Name	Parameter Type	Parameter Value
First	Enum	[(,{,[,quote,sp,NUL,~,!,`,2]
Second	Enum	[sp,string,NUL,.]
Third	Enum	[and,or,amp,pipe,sp,NUL,-,/,backslash,3]
Fourth	Enum	[sp,string,NUL,.]
Fifth	Enum	],},],quote,sp,NUL,~,%,',4]

Fig. 4. Input Model for Generating Test Search Strings

We have taken the combinatorial approach to synthesize the search strings. Fig. 4 shows the five parameters, whose values are combined to create the test strings. Each parameter is comprised of a set of strings or special characters. There are 10, 4, 10, 4, and 10 enumerated values in the respective five parameters. All possible combinations of these values create 16,000 possible test strings. In addition to the generic term "string" to represent any string, there are two special strings: "and" and "or". The term "sp" represents space and "NUL" represents an empty string. The use of "NUL" allows us to synthesize strings that can have different special characters at different positions of the synthesized strings including at the beginning and at the end of the strings. Other special characters include different types of left (*(,{,[*) and right (*),},]*) brackets, single (`, ') and double (") quotes, dot (.), ampersand (&), pipe (|), exclamation sign (*!*), hyphen (-), percent sign (*%*), slash (*/*), backslash (\), and tilde (~). These special characters are not chosen completely randomly. Since NVD allows searching for any string in CVE descriptions and other associated information within the vulnerability database (full-text search), it is conceivable that some users may construct search strings with special characters that they are looking for within the descriptions.

B. Test Set Generation

Using the input model described above, 2-way, 3-way, and 4-way covering arrays were constructed using the ACTS tool. Test set sizes and failures are shown in Table I.

TABLE I. *t*-WAY TESTS AND RESULTS, *t*=2,3,4

t	Number of tests	Number of failed tests	% of failures
2	100	12	12.0
3	999	129	12.9
4	3125	473	15.1

TABLE II. RANDOM TESTS AND RESULTS

Test set	Number of Tests	Number of failed tests	% of failures
random 1	100	13	13.0
random 2	100	17	17 .0
random 3	100	13	13 .0
random 4	100	12	12 .0
random 5	100	7	07 .0
random 6	100	7	07 .0
random 7	100	7	07 .0
random 8	100	11	11 .0
random 9	100	12	12 .0
random 10	100	12	12.0

Because random or "fuzz" testing is often used in database testing, we generated random test sets of the same size as each of the *t*-way test sets, with results as shown in Table II for $t = 2$. Tests were generated by randomly selecting a value from each of the five factors detailed above. Results varied significantly and in three cases of the 10 runs, more failures were found in the random tests than with 2-way covering arrays. This occurred because with random generation, multiple occurrences of a fault-triggering combination would appear in some test sets more than others. Following the test runs, we used the fault location tool described in [34] to locate combinations that occurred

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

in failing tests that were not also in passing tests, as described in the next section.

V. Results and Discussions

The NVD is a heavily used database, averaging approximately 7.3 million accesses per month. It was tested extensively in development, and has been in continuous use, in some form, since 1998. Its usage profile is not unlike many other large, widely used information systems. While faults have been discovered occasionally, the system continues to perform adequately for the users who rely on it. Whenever faults have been found, they have been repaired quickly and have not disrupted service. The NVD group implemented a new version of full-text search in Spring 2016 and became aware of some new issues. They fixed some of the problems with special characters. Faced with the classic SE constraints of inadequate time and resources as well as tools and techniques to easily identify all failure scenarios with the new implementation, they approached us for a systematic and thorough testing of their search functionality.

The faults identified in this paper show that even long-term operation does not guarantee eventual discovery of *all* failures. Moreover, any new feature implementation may cause a number of new failures, which is unlikely to be discovered by a traditionally designed test suite. The failure-triggering combinations found in this study are clearly "corner cases", very unusual combinations of character strings that are unlikely to occur in practical use. From a $4^2 10^3$ input configuration, we identified 49 input string combinations that result in non-timeout related failures, or roughly 0.3% of the 16,000 possible combinations in the input space as modeled. It is notable that all of these are 2-way combinations containing at least one special character.

Fault-triggering combinations can be determined using simple heuristics described in [34]. More sophisticated methods exist for fault location, e.g. [32][33], but the simple heuristics below are quick and easy to apply for this test problem. For a deterministic system, in which a given set of input values always produces the same result independent of the order of variable values, let P = {combinations in passing tests} and F = {combinations in failing tests}. The following rules were applied:

- *Elimination*: For a deterministic system, $F \setminus P$ must contain the fault-triggering combinations because if any of those in were in P, then the test would have failed.
- *Interaction level lower bound*: If all t-way tests pass, then clearly a t-way or lower strength combination did not cause the failure.
- *Interaction continuity*: For each level of t, we compute $S_t = F_t \setminus P_t$, the suspicious t-way combinations that may have triggered a failure. Because t-way tests cover all combinations of t-way or lower strength, a combination that triggered the failure in F_t must also occur in F_{t+1}, F_{t+2}, etc. So we remove any combination in S_t from S_k for any $k > t$.

Initially, $F \setminus P$, combinations in failed tests not also in passed tests, were as shown in Table III. These sets were reduced by testing each individually, resulting in 49 2-way combinations, 144 3-way combinations, and 373 4-way combinations that all triggered failures. However, a lower strength combination that triggers a failure would also produce a failure if it is contained in any higher strength combination. For example, if the 2-way combination &% triggers a failure, it will also do so in a 3-way combination &%{. Therefore, suspect 3-way combinations were removed from the 4-way suspect set and suspect 2-way combinations were removed from the 3-way and 4-way sets. As shown in Table 1, it was then possible to conclude that only 2-way combinations were responsible for all failures discovered. The complete set of failure-triggering combinations is shown in Table IV.

Table III. Interaction identification

	2-way	3-way	4-way
Initial	49	144	373
removing (t-1)-way		0	124
removing (t-2)-way			0

It is important to note that if the test goal is strong assurance that all faults have been discovered, then 3-way and 4-way testing are necessary, even though they do not discover any additional failing combinations. Without running these stronger interaction tests, we would not have been able to conclude that the 2-way combinations likely represented the complete set of faults for this input model. Any reasonable testing scheme will require that we continue testing as long as errors are being discovered. High strength covering arrays provide a stopping criterion. If no new failures are discovered after increasing t-way coverage to (t+2)-way, it is unlikely that any new faults will be found.

Table IV. Failure triggering combinations

&%	.~	or~	|4	~3
&'	/~	str&	|str	~4
&.	2&	str|	|}	~\
&4	2|	str~	|~	~and
&str	2~	{&	~%	~or
&}	3~	{|	~&	~str
&~	\~	{~	~'	~|
-~	`&	|%	~-	~}
.&	`|	|'	~.	~~
.|	`~	|.	~/	

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Text searches are among the most common tasks in information systems. Although the test procedure described in this paper addresses only this narrow problem, it is designed to be usable across the broad range of systems that require text search.

Covering arrays vs. random tests: Because *fuzz testing* is commonly used in many test situations, we compared the combinations covered by *t*-way arrays with coverage for an equal number of randomly generated tests. Fig. 5 shows a representative example for a random test set of the same size as a 3-way test set developed for the input model described in Sect. IV. The area under a curve represents the total combination coverage [35] for a given level of *t*, and the right-hand Y intercept represents the minimal coverage. For example, if variables are binary and there are one or more 2-way combinations where only 00 and 10 are covered (out of 00, 01, 10, 11), then the minimal coverage is 50%. The example test set in Fig. 5 shows approximately 95% 2-way, 84% 3-way, and 42% 4-way coverage. All 2-way combinations have at least 90% of settings covered, 3-way at least 65%, and 4-way at least 30% coverage.

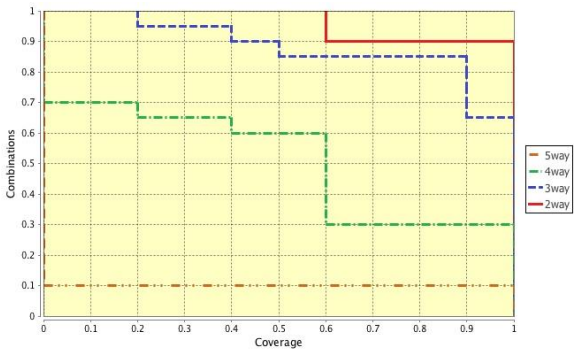


Fig. 5. Combinatorial coverage of random tests

Now consider what these coverage levels mean for assurance. With roughly 95% of 2-way combinations covered, we could expect to detect 46 or 47 of the 49 faults. So the fault detection capability of random tests, in this case, compares relatively well with covering arrays - *if we are not seeking high assurance*. Random testing falls short in two aspects for high assurance: 1) inadequate combination coverage for fault detection; and 2) inadequate coverage for a stopping criterion. For safety or mission-critical systems, finding only 95% of faults is unacceptable. Moreover, we would have no way of estimating the degree to which faults have been discovered without extending testing until the relevant input space has been covered.

Fuzz testing or other random test generation can be an efficient and appropriate means of fault discovery, but sound engineering requires a defensible method for measuring test thoroughness. Structural coverage metrics provide one set of reasonable measures - full branch or condition coverage indicates a degree to which executable code has been exercised. Measuring combinatorial coverage of tests can provide a complementary measure to structural coverage, because it shows what proportion of the input space has been included in tests. Any combination not covered by the test set is to some degree unknown territory - even with full structural coverage we do not know what the code will do with a particular combination of inputs. Similarly, extended use of a system does not guarantee that some inputs will not produce a failure, as shown by the NVD testing described in this paper. Using covering arrays makes it easy to check system response to rare inputs, to a degree that is unlikely and difficult to achieve with conventional test methods or through continuous use for many years.

VI. Future Work

Database-backed web applications usually also allow users to select advanced search options to select a subset of entries from the database that match multiple criteria. Our case study, NVD, is no exception. There is an advanced search option users can choose to narrow down their search results using many different criteria. Fig. 2. Shows an image of the NVD advanced search page. In addition to providing support for the keyword search, users can search for a specific CVE-Identifier such as CVE-2016-1234. There is option to select for a particular CWE category (e.g., CWE-94: Code Injection). Users can also choose a specific vendor or product to search for vulnerabilities associated with that vendor or product. There are two sets of date-ranges: published-date and last-modified-date that users can use to narrow their search. Additionally, there is a way to select options for different factors that define the CVSS scores of the known vulnerabilities stored in the database.

Like the keywords search, *CVE-Identifier*, *vendor*, and *product* search option allows the use of any string in the search fields. The other options are drop-down lists that users will have to choose an option from. There are 106 different CWE categories to select from in the NVD advanced search page. Similarly for CVSS version 2, users can choose from multiple options for *Severity Score Range* (SSR) of *Any*, *Low* (0-3), *Medium* (4-6), *High and Medium (4-10), and High (7-10)*. For *Attack Vector* of CVSS 2, there are options of *Any*, *Network* (N), *Adjacent* (A), and *Local* (L). There are other options available for selection for each of the other components of CVSS, which is comprised of *Access Complexity* (AC), *Authentication* (Au), *Confidentiality* (C), *Integrity* (I) and *Availability* (A). Even if we leave out the free-string search options for keywords, CVE Identifier, vendor, and product names, there are 1.45×10^{16} combinations of values for exhaustively testing the advanced search page of NVD application.

Since our goal is to look for search values and options that may lead to unexpected behavior or system failure, we

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

designed another input model with a selected subset of the parameter values for each of the search options. Fig. 6 shows the model we used to synthesize search strings to query the NVD database. In addition to utilizing the expected values for each of the advanced search options, we added an unexpected value such as "off" or "X".

Parameter Name	Parameter Type	Parameter Value
results_type	Enum	[overview,statistics,off]
cwe_id	Enum	[CWE-20,CWE-119,CWE-89,CWE-off]
pub_date_start_month	Enum	[-1,0,5,off]
pub_date_start_year	Enum	[1990,2000,2004,off]
publish_date_end_month	Enum	[6,11,12,off]
pub_date_end_year	Enum	[2007,2016,2020,off]
cvss_version2_severity	Enum	[LOW,MEDIUM,HIGH,OFF]
AV	Enum	[N,A,L,X]
AC	Enum	[L,M,H,X]
Au	Enum	[N,S,M,X]
C	Enum	[N,P,C,X]
I	Enum	[N,P,C,X]
A	Enum	[N,P,C,X]

Fig. 6. Input Model for Advanced Search Options

NVD allows direct querying of their database through the construction of search URLs. The designers left this option open for allowing programmatic search of different aspects of the information stored in the database. We utilized this feature to synthesize web search URLs combining parameter values shown in Fig. 6 and tested the NVD search engine responses. In our preliminary results, a large fraction of test cases resulted in a "Server Error" response. For example, 30 out of 33 test cases from 2-way and 767 out of 820 test cases from 4-way covering array produced an error response. The NVD developers indicate that these server errors are not necessarily bugs; rather a non-descriptive response to the end-user while processing invalid input.

It does appear that most of the unexpected parameter values result in server error response. However, not all unexpected values resulted in the error response. For example, a "-1" for *starting month* parameter results in a regular response from the application. Clearly there are some anomalies in how unexpected or invalid inputs are treated by the application. As future work, we plan to more deeply investigate the resiliency of NVD system against unexpected parameter values for advanced search options. We also plan to research the coverage we can gain from applying combinatorial testing approach over the 'Advanced Search's parameter input space. It would be interesting and useful to determine the combination factors that can cause 'Advanced Search' to fail. Initial test results have already revealed that certain valid CWE-category search can also cause failures while combined with other valid parameter values, which is something the application developers did not anticipate.

VII. Conclusions

We investigated the application of combinatorial testing to string text searches in the US National Vulnerability Database, a system that is accessed more than 70 million times a year. The current software build is operational 24 hours a day. Our testing and analysis revealed 49 inputs that produced server errors in the current build. These inputs were 2-way combinations of special characters and strings, and test cases built from 2-way through 4-way covering arrays demonstrated that no other combinations beyond these 49 resulted in the server error response. This result demonstrates the effectiveness of combinatorial methods for detecting and determining the full extent of rare faults.

The test procedure described in this paper addresses a specific test problem. It can be applied with little or no change to many systems that incorporate text searches. Text search is an essential component in systems within nearly all industries, and some are safety or mission-critical. Applying test methods such as those described in this paper can help to remove rare faults that could result in significant failures in operation.

Equally important, the methods described here provide a defensible criterion for test completion. Because covering arrays include all *t*-way factor combinations, we can show that the entire input space has been covered up to whatever *t*-way combinations are used. In contrast, "fuzz testing" or other conventional methods do not include measures of the input space that has been tested, and often rely on a "more is better" heuristic without an ability to measure completeness. Using covering arrays, or measuring combinatorial coverage of random tests, provides a sound test engineering method with defensible, quantitative measures of test completeness.

Acknowledgements

The authors would like to thank the NIST NVD group at for their assistance with this work. This study was supported by NIST ITL Grant 70NANB17H035.

Disclaimer: *Products may be identified in this document, but identification does not imply recommendation or endorsement by NIST, nor that the products identified are necessarily the best available for the purpose*

References

[1] H. Zhu, P. A. V. Hall, and J. H. R. May, "Software unit test coverage and adequacy," ACM Computing Surveys, vol. 29, no. 4, pp. 366–427, 1997.

[2] M. Heusser, "6 software development lessons from healthcare.gov's failed launch," CIO, November 2013.

[3] D. Doherty, "Team obama never finished testing healthcare.gov before launching it," CBS News, November 2013.

[4] R. Kuhn, R. Kacker, Y. Lei, and J. Hunter, "Combinatorial software testing," Computer, vol. 42, no. 8, pp. 94–96, Aug 2009.

[5] L. S. Ghandehari, J. Czerwonka, Y. Lei, S. Shafiee, R. Kacker, and R. Kuhn, "An empirical comparison of combinatorial and random testing," 2014 IEEE Seventh

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Intl Conf on Software Testing, Verification and Validation Workshops, March 2014, pp. 68–77.

[6] K. Haller, "The test data challenge for database-driven applications," Third Intl Workshop on Testing Database Systems, ACM, 2010, pp. 6:1– 6:6.

[7] M. J. Su´arez-Cabal and J. Tuya, "Using an sql coverage measurement for testing database applica-tions," SIGSOFT Softw. Eng. Notes, vol. 29, no. 6, pp. 253–262,2004.http://doi.acm.org/10.1145/1041685.1029929

[8] M. Emmi, R. Majumdar, and K. Sen, "Dynamic test input generation for database applications," in Proceedings of the 2007 Intl Symp on Software Testing and Analysis, ser. ISSTA '07. New York, NY, USA: ACM, 2007, pp. 151–162.

[9] K. Taneja, Y. Zhang, and T. Xie, "Moda: Automated test generation for database applications via mock objects," IEEE/ACM Intl Conf on Automated Software Eng., New York, NY, USA: ACM, 2010, pp. 289–292.

[10] A. Bertolino, "Software testing research: Achievements, challenges, dreams," in Proc. of ICSE Future of Software Engineering (FOSE), 2007, pp. 85–103.

[11] D. Willmor and S. M. Embury, "An intensional approach to the specification of test cases for database applications," 28th Intl Conf on Software Engineering, New York, NY, USA: ACM, 2006, pp. 102–111

[12] NIST. (2007) National vulnerability database (nvd). [Online]. Available: https://nvd.nist.gov/

[13] MITRE/CVE. (2003) https://cve.mitre.org

[14] NIST/ CVSS. (2012) https://nvd.nist.gov/cvss.cfm

[15] Tsumura, K., et al., April. Pairwise coverage-based testing with selected elements in a query for database applications. *Software Testing, Verification and Validation Workshops (ICSTW), 2016 IEEE Ninth Intl Conf on* (pp. 92-101). IEEE.

[16] Bozic, J., Simos, D.E. and Wotawa, F., 2014, May. Attack pattern-based combinatorial testing. *9th Intl Wrkshp on Automation of Software Test* (pp. 1-7). ACM.

[17] Garn, B., Kapsalis, I., Simos, D.E. and Winkler, S., On the applicability of combinatorial testing to web application security testing: a case study. *2014 Workshp Joining AcadeMiA and Industry Contributions to Test Automation and Model-Based Testing* (pp. 16-21).

[18] Bozic, J., Garn, B., Simos, D.E. and Wotawa, F., April. Evaluation of the IPO-family algorithms for test case generation in web security testing. In *Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth Intl Conf on* (pp. 1-10). IEEE.

[19] Ratliff, Z.B., Kuhn, D.R., Kacker, R.N., Lei, Y. and Trivedi, K.S., The Relationship between Software Bug Type and Number of Factors Involved in Failures. In *Software Reliability Engineering Workshops (ISSREW), 2016 IEEE Intl Symp on* (pp. 119-124). IEEE.

[20] K. Haller, "The test data challenge for database-driven applications," Third Intl Workshop on Testing DatabaseSystems, ser. DBTest '10. New York, NY, USA: ACM, 2010, pp. 6:1– 6:6.

[21] Ghandehari, L.S., Czerwonka, J., Lei, Y., Shafiee, S., Kacker, R. and Kuhn, R., 2014, March. An empirical comparison of combinatorial and random testing. In *Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh Intl Conf on* (pp. 68-77). IEEE.

[22] Schroeder, P. J., Bolaki, P., & Gopu, V. (2004, August). Comparing the fault detection effectiveness of n-way and random test suites. In *Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 Intl Symp on* (pp. 49-59). IEEE.

[23] Ellims, M., Ince, D. and Petre, M., 2008, September. The effectiveness of t-way test data generation. In *Intl Conf on Computer Safety, Reliability, and Security* (pp. 16-29). Springer Berlin Heidelberg.

[24] Vilkomir, S., Starov, O. and Bhambroo, R., 2013, March. Evaluation of t-wise approach for testing logical expressions in software. In *Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth Intl Conf on* (pp. 249-256). IEEE.

[25] Ballance, W.A., Vilkomir, S. and Jenkins, W., April. Effectiveness of pair-wise testing for software with boolean inputs. *Software Testing, Verification and Validation, 2012 IEEE Fifth Intl Conf* (pp. 580-586)

[26] Kobayashi, N., Tsuchiya, T. and Kikuno, T., 2001, July. Applicability of non-specification-based approaches to logic testing for software. In *Dependable Systems and Networks, 2001. DSN 2001.* (pp. 337-346). IEEE.

[27] Bell, K.Z. and Vouk, M.A., 2005, December. On effectiveness of pairwise methodology for testing network-centric software. In *Information and Communications Technology, 2005. Enabling Technologies for the New Knowledge Society: ITI 3rd Intl Conf on* (pp. 221-235). IEEE.

[28] Bryce, R.C. and Colbourn, C.J., 2006. Prioritized interaction testing for pair-wise coverage with seeding and constraints *Inf.Software Tech.*, *48*(10), pp.960-970.

[29] Ghandehari, L.S., Czerwonka, J., Lei, Y., Shafiee, S., Kacker, R. and Kuhn, R., An empirical comparison of combinatorial and random testing. *Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh Intl Conf on* (pp. 68-77). IEEE.

[30] Bartholomew, R. (2013, May). An industry proof-of-concept demonstration of automated combinatorial test. In *Automation of Software Test (AST), 2013 8th Intl Workshop on* (pp. 118-124). IEEE.

[31] Borazjany, Mehra N., et al. "An input space modeling methodology for combinatorial testing." *Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth Intl Conf on*. IEEE, 2013.

[32] Colbourn, C. J., & McClary, D. W. (2008). Locating and detecting arrays for interaction faults. *Journal of combinatorial optimization*, *15*(1), 17-48.

[33] Wang, Z., Xu, B., Chen, L., & Xu, L. (2010, July). Adaptive interaction fault location based on combinatorial testing. In *Quality Software (QSIC), 2010 10th Intl Conf on* (pp. 495-502). IEEE.

[34] Kuhn, D. R., Kacker, R. N., & Lei, Y. (2010). SP 800-142. Practical Combinatorial Testing.

[35] Kuhn, D. R., Kacker, R. N., & Lei, Y. (2016). Measuring and specifying combinatorial coverage of test input configurations. *Innovations in Systems and Software Engineering*, *12*(4), 249-261.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Material Measurement Laboratory

Work of researchers at professional conferences as reported in Fiscal Year 2017

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

USING DESIGN OF EXPERIMENTS IN FINITE ELEMENT MODELING TO IDENTIFY CRITICAL VARIABLES FOR LASER POWDER BED FUSION

Li Ma, Jeffrey Fong, Brandon Lane, Shawn Moylan, James Filliben, Alan Heckert, and Lyle Levine

National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899

REVIEWED

Abstract

Input of accurate material and simulation parameters is critical for accurate predictions in Laser Powder Bed Fusion (L-PBF) Finite Element Analysis (FEA). It is challenging and resource consuming to run experiments that measure and control all possible material properties and process parameters. In this research, we developed a 3-dimensional thermal L-PBF FEA model for a single track laser scan on one layer of metal powder above a solid metal substrate. We applied a design of experiments (DOE) approach which varies simulation parameters to identify critical variables in L-PBF. DOE is an exploratory tool for examining a large number of factors and alternative modeling approaches. It also determines which approaches can best predict L-PBF process performance.

1. Introduction

Laser powder bed fusion (L-PBF) is an additive manufacturing (AM) technology for the fabrication of near net shaped parts directly from computer-aided design (CAD) data by sequentially melting layers of metal powder with a laser source. L-PBF is one of the most promising additive manufacturing processes because it provides better surface and geometric part quality compared to other metal AM technologies. However, the highly localized laser power input leads to extremely high local temperature gradients. As a result, significant residual stresses, distortion, unique microstructures, and defects may occur within a workpiece. A round robin comparison of mechanical properties [1] found that the quality and properties of deposits can vary significantly even when all producers are using the same materials, processing parameters, and, in some cases, even when the same type of L-PBF machine is used.

L-PBF finite element modeling plays an important role in understanding the L-PBF process, predicting optimal fabrication strategies, and qualifying fabricated parts based on those strategies. Accurate temperature prediction from computational thermal modeling is also critical for modeling microstructure evolution and residual stresses. Although several thermal finite element analysis (FEA) models appear in the literature [2-14], significant challenges remain to construction of accurate FEA simulations of the L-PBF process. Input of accurate material and simulation parameters is critical for accurate prediction of process signatures, such as peak temperature, melting pool size, etc. The measurement and control of all possible material properties and processing parameters is challenging and resource consuming. Therefore, a computational design of experiments (DOE) approach was undertaken to simplify this task.

Filliben, James; Fong, Jeffrey; Heckert, Nathanael; Lane, Brandon; Levine, Lyle; Ma, Li; Moylan, Shawn.
"Using DOE in Finite Element Modeling to Identify Critical Variables in Laser Powder Bed Fusion."
Paper presented at 2015 Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference,
Austin, TX. August 10, 2015 - August 12, 2015.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

In this research, we developed 3-dimensional thermal FEA models of L-PBF process. These FEA L-PBF thermal models incorporate a continuous moving heat source, phase changes, and powder thermal property changes after melting. A single track laser scan on one layer of metal powder above a solid metal substrate was modeled. A computational DOE approach was used that varied simulation parameters to identify the critical variables for accurate representation of the L-PBF process.

2. Computational Design of Experiments

As discussed above, to explore the dominant factors contributing to the uncertainty of the L-PBF process, we applied a computational DOE approach. The results from the FEA models with a range of processing parameters and materials properties provide the evaluation input for the DOE.

2.1. Model description

Using the commercial FEA code ABAQUS[1] [15], a non-linear, transient, thermal model was designed and executed to obtain the global temperature history generated during a single AM laser scan. The mesh design is shown in Fig. 1. The specimen is a solid substrate with one layer of powder. The dimensions of the specimen are 6 mm (length) × 1.4 mm (width) × 0.6 mm (thickness). The powder layer thickness is 37 μm. A single-track laser scan across the metal powder layer was modeled. To reduce computation time, the elements that interact with the laser beam are finely meshed with six hexahedral elements within the diameter of the laser, and a coarse mesh is used for the surrounding loose powder and substrate. Within the fine mesh, the element size is 16.7 μm (length) × 16.7 μm (width) × 12.3 μm (thickness).

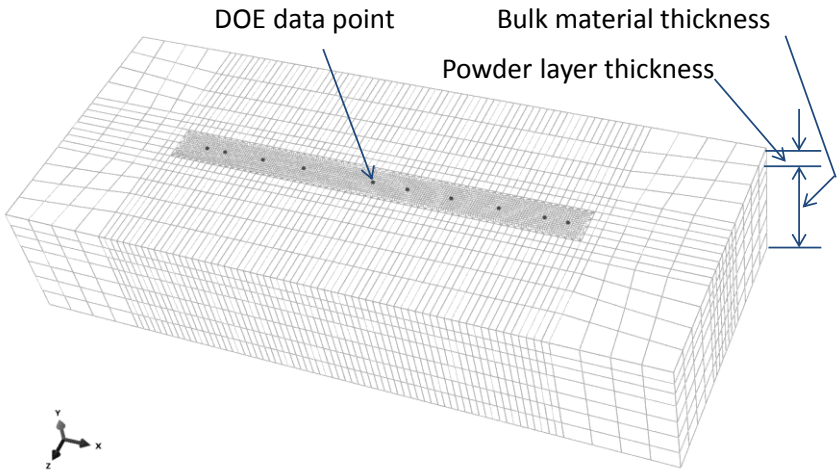


Figure1: Finite element one track thermal model mesh. The dark dots are the points used for the temperature history output.

[1] Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

2.2. Thermal modeling

The heat conduction in the L-PBF process was modeled using the Fourier heat conduction equation given by Carslaw and Jaeger [16]:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k \frac{\partial T}{\partial z}\right) + Q \quad (1)$$

where ρ is the material density; c is the specific heat capacity; k is the thermal conductivity; T is the temperature; t is the interaction time, and Q is the internal heat.

The initial condition assumed a uniform temperature distribution throughout the specimen at time $t = 0$, which can be expressed as

$$T(x, y, z, 0) = T_0 \quad (2)$$

where T_0 is the preheat temperature taken as 353 K (80 °C).

The boundary condition on the top surface includes the input heat flux, surface convection, and radiation that follow the equation:

$$(-k\nabla T) \cdot \hat{n} = q_s + h(T - T_e) + \varepsilon_\theta \sigma\left(T^4 - {T_e}^4\right) \quad (3)$$

where q_s represents the laser heat input, h is the convection heat transfer coefficient, ε_θ is the thermal radiation coefficient, σ is the Stefan-Boltzmann constant, and T_e is the ambient temperature. An adiabatic boundary condition was applied to all the surfaces except the top surface of the specimen.

2.4. Modeling the moving heat flux

The laser in this study is the continuous Ytterbium fiber diode laser (wavelength = 1.064 μm) that is widely used in actual L-PBF processes. A user subroutine was developed to simulate the characteristics of the heat flux of the laser onto the sample surface. The surface heat flux of the laser beam is modeled as a Gaussian distribution [7]:

$$q_s = \frac{2AP}{\pi r_b} exp\left(\frac{-2r^2}{r_b^2}\right), \quad (4)$$

where P is the laser power, A is the absorption coefficient of the powder layer, r is the radial distance relative to the center of laser beam, and r_b is the radius of the laser beam, which is 50 μm in our simulations.

2.5. Materials properties

Inconel 625 was used in this study because: a) it is widely used in the L-PBF process, and b) it was the material used in the L-BPF round robin tests [1].

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The powder packing ratio, φ, is a function of the local density of the powder, ρ_{powder}, and the density of the solid material, ρ_{bulk}:

$$\varphi = \frac{\rho_{powder}}{\rho_{bulk}}. \quad (5)$$

In our FEA model, the initial powder density linearly increases to the bulk density when the temperature is above the solidus temperature, T_s, and below the liquidus temperature, T_l, on the Inconel 625 phase diagram [20, 21]. The initial powder-state elements are irreversibly changed to bulk-state elements when the temperature exceeds T_l. Consequently, the density and thermal conductivity of the powder bed are treated as a function of temperature and a melt-state variable which records if the powder has experienced the first melting point. These changes are performed by the ABAQUS user subroutine and ρ_{powder} is defined as:

$$\rho_{powder} = \begin{cases} \varphi\rho_{bulk}(T), \ T \le T_s, \ Before\ first\ melt \\ \varphi\rho_{bulk}(T_s) + \frac{\rho_{bulk}(T_s) - \varphi\rho_{bulk}(T_s)}{T_l - T_s}(T - T_s), \ T_l < T < T_s, \ First\ melt. \\ \rho_{bulk}(T), \ T \ge T_l, \ First\ melt \\ \rho_{bulk}(T), \ After\ first\ melt \end{cases} \quad (6)$$

From prior research [17, 18], the effective thermal conductivity of a powder bed depends not only on the conductivity of the bulk material, but also on the packing fraction, the particle size distribution, the particle morphology, and the thermal conductivity of the surrounding gas. It was found that the thermal conductivity of the powder, k_{powder}, is much smaller than that of the bulk material at room temperature [17]. In these simulations, the thermal conductivity of the powder is not directly connected to the powder packing ratio and $k_{powder}(T)$ ranged from 1.0 W/mK to 3.0 W/mK [19] before the first melt. The effective thermal conductivity of the powder $k_{powder,eff}$ is defined as

$$k_{powder,eff} = \begin{cases} k_{powder}(T), \ T \le T_s, \ Before\ first\ melt \\ k_{powder}(T_s) + \frac{k_{bulk}(T_s) - k_{powder}(T_s)}{T_l - T_s}(T - T_s), \ T_l < T < T_s, \ First\ melt. \\ k_{bulk}(T), \ T \ge T_l, \ First\ melt \\ k_{bulk}(T), \ After\ first\ melt \end{cases} \quad (7)$$

where k_{bulk} is the thermal conductivity of the bulk material. The latent heat was added when the temperature was between the solidus and liquidus temperatures. The temperature-dependent bulk material density and specific heat were calculated from a Scheil simulation for the nominal IN625 composition and using the TCNI6 thermodynamic database [20] within the Thermo-Calc software [21].

2.6. Design of experiments

In the L-PBF process, there are dozens of factors that may influence the quality of a final manufactured part [22]. It is challenging and expensive to measure and control all possible material properties and process parameters. For example, if 50 different factors interact with

each other, a proper uncertainty budget should consider, at least, the roughly 1200 2-term interactions as well as the nearly 20,000 3-term interactions formed from combinations of the original 50 parameters. Investigating all of these interactions experimentally is unfeasible. The task of determining the importance of such a large number of factors can be greatly simplified by using *a priori* knowledge of the processing, experimental experience, computation modeling experience, and statistical/computational DOE methods.

To demonstrate this approach, ten factors thought to be important were selected for DOE analysis. These ten factors include both processing and material properties parameters. The choice of factors is guided by prior research, experience on processing quality control, and computational modeling. An exhaustive screening design would likely consider many more factors and a corresponding increase in the number of simulation runs. Since the goal of this preliminary work is to identify dominant factors, and not to fully characterize the response function, we choose a two-level screening design. As the ranges of most factors in the L-PBF process are not available, the values of the two levels (high and low, or + and -) for each of the factors come from our experience, just for demonstration.

The factor and level combinations needed for the screening were drawn from standard tables [23]. In this case, $2_{III}^{(10-6)}$ fractional factorial design was chosen. In this notation, the '2' indicates a two-level design (two possible values for each input parameter), the "10" indicates that ten factors or parameters are considered and the "*III*" reveals that this design is resolution three, which means that the main effects of the ten variables are not confounded with any 2-term interactions. Some confounding of 2-term interactions with each other is present, however. This design requires $2^{(10-6)} = 2^4 = 16$ simulation runs. Table 1 listed the input parameters for the 16 runs used in this research.

Table1: List of input parameters and a resolution III fractional factorial orthogonal design for a 10-factor, 16-run numerical FEM experiment

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
Factor Symbol	*E*	*hc*	*Ti*	*Ab*	*Rho*	*Cp*	*k*	*Dp*	*P*	*v*
Factor Meaning	Emmisivity	Convection	Preheat Temperature	Absorption	Density	Specific heat	Thermal Conductivity	Powder Packing Ratio	Laser Power	Scanning Speed
Base Run (00)	0.37	0.05	353	0.12	*Rho (T)*	*Cp (T)*	*k (T)*	50	195	800
Factor Unit		*W/K/m^2*	*K*		*kg/mm^3*	*J/kgK*	*W/mK*	*%*	*W*	*mm/s*
+/- variation	10%	10%	1%	0.5%	1%	3%	3%	10%	2.5%	1.5%
Run No.(01)	-	-	-	-	-	-	-	-	+	+
Run No.(02)	+	-	-	-	+	-	+	+	-	-
Run No.(03)	-	+	-	-	+	+	-	+	-	-
Run No.(04)	+	+	-	-	-	+	+	-	+	+
Run No.(05)	-	-	+	-	+	+	+	-	-	+
Run No.(06)	+	-	+	-	-	+	-	+	+	-
Run No.(07)	-	+	+	-	-	-	+	+	+	-
Run No.(08)	+	+	+	-	+	-	-	-	-	+
Run No.(09)	-	-	-	+	-	+	+	+	-	+
Run No.(10)	+	-	-	+	+	+	-	-	+	-
Run No.(11)	-	+	-	+	+	-	+	-	+	-
Run No.(12)	+	+	-	+	-	-	-	+	-	+
Run No.(13)	-	-	+	+	+	-	-	+	+	+
Run No.(14)	+	-	+	+	-	-	+	-	-	-
Run No.(15)	-	+	+	+	-	+	-	-	-	-
Run No.(16)	+	+	+	+	+	+	+	+	+	+

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

We note that a k-factor, 2-level orthogonal design, such as the one used in this study, has a balanced number of settings for each factor, and for every pair of factors. Such balance yields many advantages, including: 1) coverage and robustness: the design points provide coverage across the entire k-space of factors, thus yielding robust effect estimates with minimal bias; 2) uncertainty-reduction: each factor effect estimate uses all n observations, thus making the uncertainty for each estimate is as small as possible; 3) superiority over 1-factor-at-a-time experiments: orthogonal designs minimize factor confounding/contamination and maximize (if possible) the ability to estimate interactions; 4) hypothesis testing: if a factor is in fact significant in reality, then our ability to carry out a hypothesis test and conclude the factor is "significant" is maximized; and 5) simplified least squares: the resulting factor effect estimates are least squares equivalent and simplify to (average Y at high setting) – (average Y at low setting).

3. Results

3.1. FEA temperature profile

Figure 2 shows the center point on the scan surface (as shown in Fig. 1) temperature as a function of time for all 16 computational DOE runs. It can be seen that the temperature profile shows a clear transition between the solidus and liquidus temperatures because of the latent heat. For simplicity in this first attempt, we selected the peak temperature of each profile as the DOE input data. In the future, the geometry of the melt pool can be used for a more physically useful parameter.

3.2. Design of Experiments Result

From the FEA model of a single laser scan track using the parameters specified in Table 1 for each of the 16 runs, we obtained the peak temperature at the center point of the specimen top surface during the scanning process. We then conducted a sensitivity analysis using these 16 runs plus a center point (the base design solution) computational experiment, using a computer code written in DATAPLOT [24].

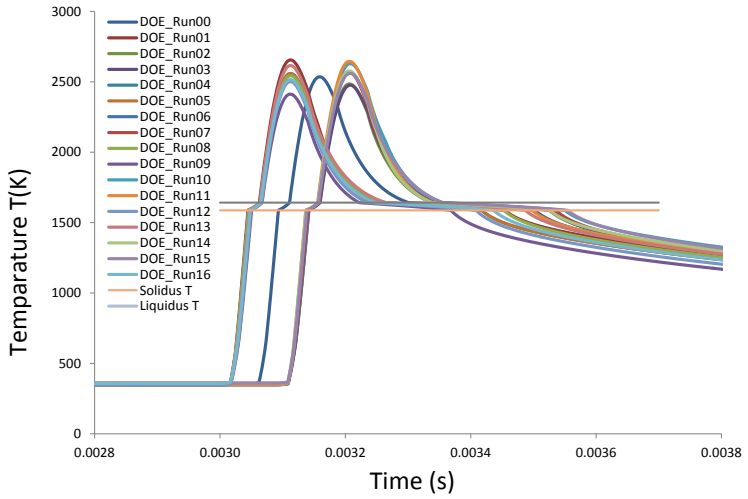


Figure 2: The temperature at the center point of the scan surface (as shown in Fig. 1) as a function of time for all 16 computational DOE runs.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

3.2.1 Effects order

One useful tool for quickly visualizing dominant factors is to plot the main effect as shown in Figure 3. When displayed in this format, factors that have a large impact on the peak temperature appear as line segments with large slope. The slope directions denote the peak temperature change direction with the individual factors. In Fig. 3, we observed that the laser power (X9) and material specific heat (X6) have the largest effects, with the laser power being statistically significant at the 95 % percent level. As expected, increasing the laser power will increase the peak temperature while the materials with higher specific heat will generate lower peak temperature.

Figure 4 displays the order of the ten factors affecting the peak temperature. It can be seen that besides laser power (X9) and specific heat (X6), the other dominant factors include the laser scan speed (X10) and the powder packing ratio (X8). Each of these four factors produces a temperature response ≥ 2 %. Lower response factors (≈ 1 %) include the thermal conductivity (X7) and density (X5). Convection has the least impact on the peak temperature.

3.3.3. Significance and limitations of the computational DOE approach as a tool for identifying critical variables in L-PBF

DOE-based sensitivity analysis can play an important role in quality control for additive manufacturing of parts. However, this approach is limited in the sense that it requires the user to exercise judgement in selecting the appropriate number of the parameters for implementation. In case of doubt, one can, nevertheless, try several schemes to obtain reasonable results.

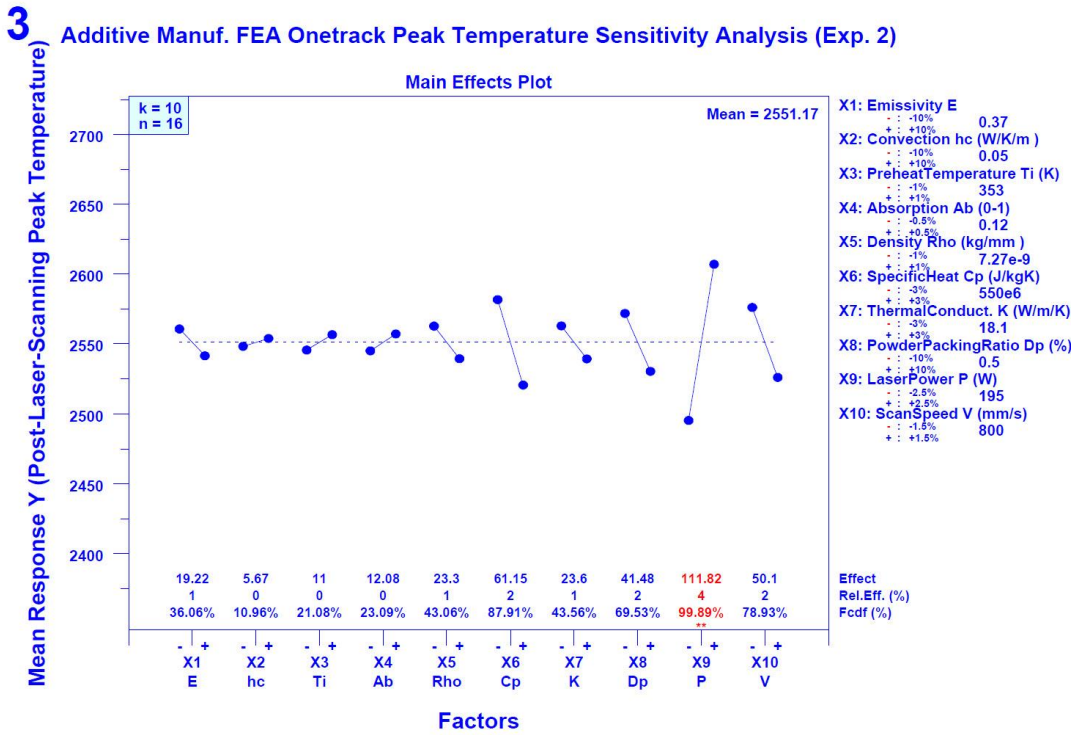


Figure 3: Main effects plot of the 10-factor, 16-run 2-level, fractional factorial orthogonal DOE.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Meanwhile, the current preliminary thermal modeling neglects important factors such as the powder shape and geometry, shrinkage, liquid solid interactions, the dependence of powder thermal conductivity on the powder packing ratio, etc. Also, the processes of vaporization and splattering were neglected. Eventually, all of these factors must be evaluated to determine their role in producing reliable and consistent AM-manufactured parts.

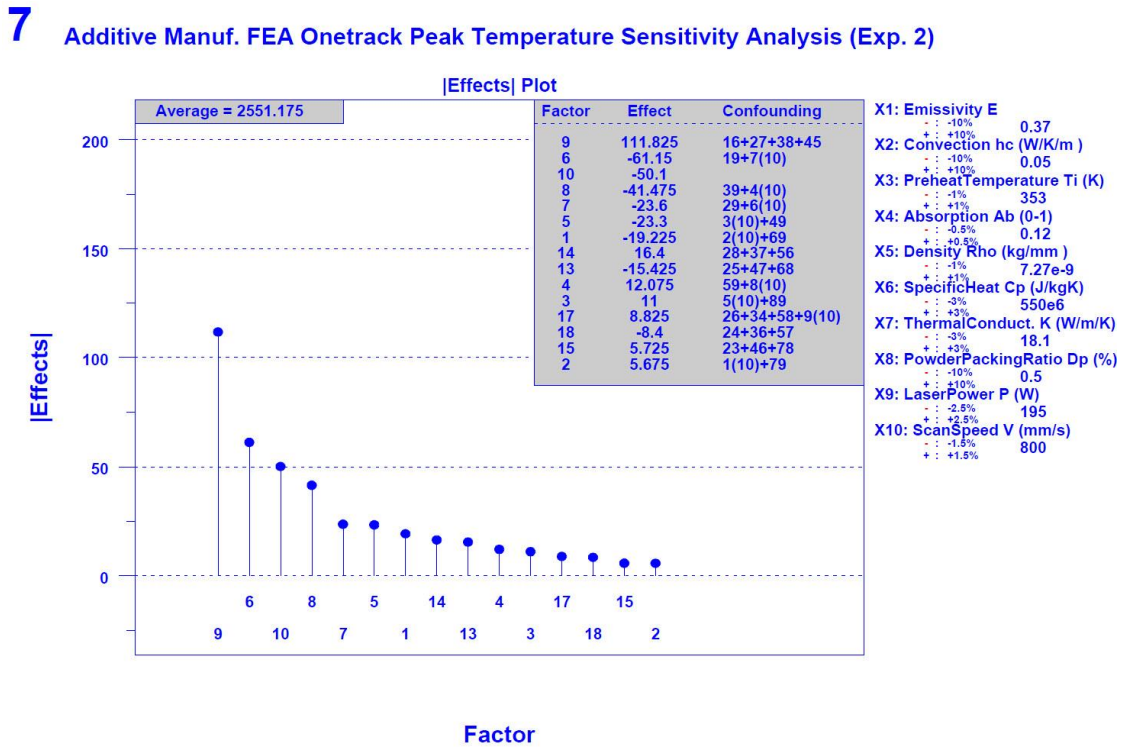


Figure 4: Ranking of effects plot of the 10-factor, 16-run 2-level, fractional factorial orthogonal DOE

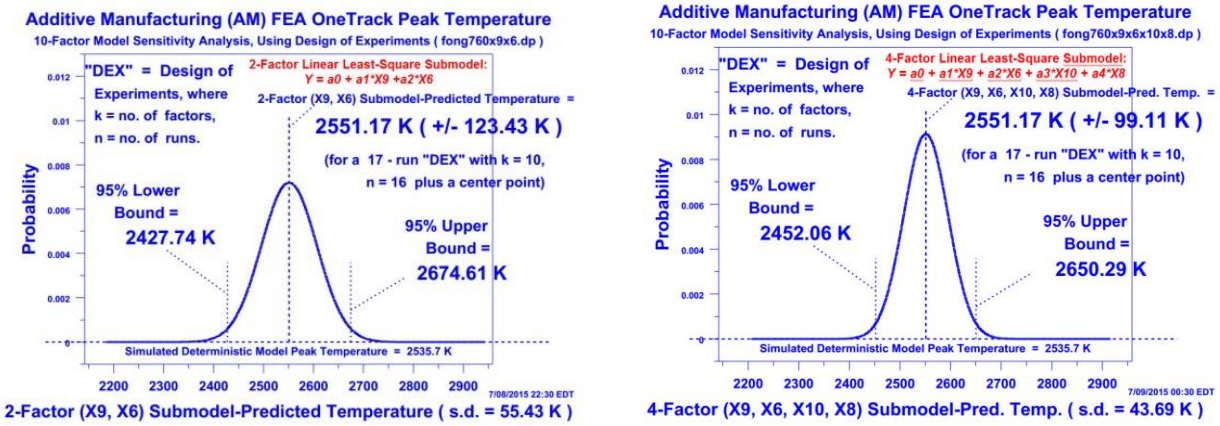


Figure 5: Uncertainty estimation based on (a) 2-Factor and (b) 4-Factor linear least-square submodel.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

4. Concluding Remarks

We characterized FEA thermal modeling of a single track L-PBF process using a 10-factor, 16-run, 2-level, fractional factorial orthogonal design of experiments. We obtained the order of dominant factors affecting the IN625 single track peak temperature as (1) laser power, (2) specific heat, (3) laser scan speed, and (4) powder packing ratio; all of these factors affected the peak temperature by ≥ 2 %. Processing parameters (laser power and scanning speed) and material properties (specific heat and powder packing ratio) both impact the uncertainty quantification and the AM part quality.

Our computational design of experiments method provides an exploratory tool for examining a large number of factors and alternative modeling approaches, allowing us to determine which approaches can best predict AM process performance. The largest potential impact of this work is to determine what process parameters and material properties most affect the quality of an AM-manufactured part. This will allow AM process experts to concentrate their efforts on those factors that have the largest impact.

Acknowledgements

The authors gratefully acknowledge the valuable discussions on the modeling and DOE with Alkan Donmez and Richard E. Ricker from NIST, and IN625 thermal properties with Carelyn E. Campbell, William Boettinger, and Sudha Cheruvathur from NIST.

References

[1] EWI submitted to NIST, "Volume 1: Development and Measurement Analysis of Design Data for Laser Powder Bed Fusion Additive Manufacturing of Nickel Alloy 625 Final Technical Report", August 28, 2014. http://ewi.org/eto/wp-content/uploads/2015/04/70NANB12H264_Final_Tech_Report_EWI_53776GTH_Distribution_Vol_1.pdf

[2] Shiomi, M., Yoshidome, A., Abe, F., and Osakada, K., "Finite element analysis of melting and solidifying processes in laser rapid prototyping of metallic powders", International Journal of Machine Tools & Manufacture, 39, 237-252, 1999.

[3] Matsumoto, M., Shiomi, M., Osakada, K., and Abe, F., "Finite element analysis of single layer forming on metallic powder bed in rapid prototyping by selective laser processing", International Journal of Machine Tools & Manufacture, 42, 61–67, 2002.

[4] Ameer, K. I., Derby, B., and Withers, P. J., "Thermal and Residual Stress Modelling of the Selective Laser Sintering Process", Materials Research Society,758, 47-52, 2003.

[5] Kolossov, S., and Boillat, E., "3D FE simulation for temperature evolution in the selective laser sintering process", International Journal of Machine Tools and Manufacture, 44 (2-3),117-123, 2004.

[6] Patil, R. B., and Yadava, V., "Finite element analysis of temperature distribution in single metallic powder layer during metal laser sintering", International Journal of Machine Tools and Manufacture 47(7-8): 1069-1080, 2007.

[7] Roberts, I. A., Wang, C. J., Esterlein, R., Stanford, M., and Mynors, D. J., "A three-dimensional finite element analysis of the temperature field during laser melting of metal powders in additive layer manufacturing", International Journal of Machine Tools and Manufacture, Vol. 49 iss:12 pp. 916-923, 2009.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

[8] Dong, L., Makradi, A., Ahzi, S., and Remond, Y., "Three-dimensional transient finite element analysis of the selective laser sintering process", Journal of Materials Processing Technology, 209, 700–706, 2009.
[9] Zhang, D. Q., Cai, Q. Z., Liu, J. H., Zhang, L., and Li, R. D., "Select laser melting of W–Ni–Fe powders: simulation and experimental study", The International Journal of Advanced Manufacturing Technology, 51(5-8), 649-658, 2010.
[10] Li, C., Y. Wang, et al. "Three-dimensional finite element analysis of temperatures and stresses in wide-band laser surface melting processing", Materials & Design, 31(7): 3366-3373, 2010.
[11] Song, B., Dong, S., Lao, H., and Coddet, C., "Process parameter selection for selective laser melting of Ti6Al4V based on temperature distribution simulation and experimental sintering", The International Journal of Advanced Manufacturing Technology, 61(9-12): 967-974, 2012.
[12] Shuai, C., Feng, P., Gao, C., Zhou, Y., and Peng, S., "Simulation of dynamic temperature field during selective laser sintering of ceramic powder", Mathematical and Computer Modelling of Dynamical Systems, 19(1), 1-11, 2013.
[13] Yin, J., Zhu, H., Ke, W., Dai, C., and Zuo, D., "Simulation of temperature distribution in single metallic powder layer for laser micro-sintering", Computational Materials Science 53(1), 333-339, 2012.
[14] Hodge, N. E., Ferencz, R. M., and Solberg, J. M., "Implementation of a thermomechanical model for the simulation of selective laser melting", Computational Mechanics, 54(1), 33-51, 2014.
[15] Abaqus, Theory and User's manual, version 6.13, Dassault Systèmes Simulia Corp., Providence, RI., USA, 2013.
[16] Carslaw, H.S., and Jaeger, J.C., "Conduction of Heat in Solids", Oxford University Press, Amen House, London E.C.4, 2005.
[17] Rombouts, M., Froyen, L., Gusarov, A. V., Bentefour, E. H., and Glorieux, C., "Photopyroelectric measurement of thermal conductivity of metallic powders", Journal of Applied Physics, 97(2), 013534, 2005.
[18] Alkahari, M. R., Furumoto, T., Ueda, T., Hosokawa, A., Tanaka, R., and Abdul Aziz, M.S., "Thermal conductivity of metal powder and consolidated material fabricated via selective laser melting", Key Engineering Materials, 523-524, 244-249, 2012.
[19] Childs, T., Hauser, C., and Badrossamay, M., "Selective laser sintering (melting) of stainless and tool steel powders: experiments and modelling, Proc Inst Mech Eng, Part B: Journal Engineering Manufacturing, 219(4), 339–57, 2005.
[20] TC Ni-based Superalloys Database, version 6; Thermo-Calc Software, Stockholm, Sweden, 2013.
[21] Thermo-Calc, version 3.1, Thermo-Calc software AB, Stockholm, Sweden, 2014.
[22] Mani, M., Lane, B., Alkan, M. A., Feng, S., Moylan, S. and Fesperman, R., " Measurement science needs for real-time control of additive manufacturing powder bed fusion processes NISTIR 8036", National Institute of Standards and Technology, 2015.
[23] Box, G. E. P., Hunter, H. G., and Hunter, J. S., Statistics for Experimenters: An introduction to design, data analysis, and model building, New York, Wiley, 1978.
[24] Filliben, J. J., and Heckert, N. A., DATAPLOT: A Statistical Data Analysis Software System, a public domain software released by NIST, Gaithersburg, MD 20899, http://www.itl.nist.gov/div898/software/dataplot.html, 2002.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Estimation of waveform state levels and uncertainties using the histogram and shorth methods

Mark Bieler[1] and Nicholas Paulter[2]

[1]Physikalisch-Technische Bundesanstalt, Bundesallee 100, D-38116 Braunschweig, Germany
mark.bieler@ptb.de

[2]National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA
paulter@nist.gov

***Abstract* — State level calculations are fundamental to extract parameters of a time-domain waveform. Here we compare state level calculations using two different estimators, the histogram mode method and the shorth method. Our results show that both methods yield accurate results. The conclusions are applicable to any waveform ranging from slow geological to ultrafast optical events.**

***Index Terms* — documentary standards, histogram, mode, shorth, state level, uncertainty, waveform.**

I. Introduction

The International Electrotechnical Commission (IEC) and the Institute of Electrical and Electronics Engineers (IEEE) publish documentary standards [1,2] for waveform parameters that are used to describe the characteristics of a waveform. These standards also contain suggested methods of computing the values of these waveform parameters. A documentary standard for the computation of waveform parameter uncertainty was recently started [3] to complement the existing waveform parameter standards.

The purpose of this manuscript is to support these standards by comparing two methods of computing state level (*level*(s), where "s" refers to a state) and its uncertainty. *level*(s) is common to almost all of the other important waveform parameters. Accordingly, the measurement uncertainty of *level*(s) affects the measurement uncertainty of all those other waveform parameters. We chose two different estimators for *level*(s), the histogram mode and the shorth for our study. The mode is the bin from a histogram of waveform values that has the largest number of occurrences of waveform values. The *level*(s) is equal to the middle of the range of the mode bin. The shorth is a specific nondecreasing sequence of waveform values comprising a certain fraction of the waveform values in the state. The *level*(s) is computed as the average value of this shorth sequence. While the histogram mode is one of the most commonly used estimators in industry, the shorth estimator is considered as an alternative to the mode [4].

II. Waveform Construction

Our analysis is based on simulated waveforms, which have been constructed as follows. First, we use a waveform containing 10 000 elements (samples) that are equally spaced in time and based on the convolution of an impulse with a Butterworth or a Chebyshev filter. The advantage of using these functions is that they approach a fixed value equal to zero at later times. This waveform type was replicated 100 times. Next, we added multiplicative and additive noise to each of the 100 waveforms. The additive noise was determined independently for each waveform sample (with mean equal to zero and a standard deviation ranging between 1% and 10% of the waveform maximum). The multiplicative noise was determined independently for each of the 100 waveforms (with mean equal to one and a standard deviation ranging between 1% and 10%). The 100 noisy waveforms and their average were taken for the state level analysis of multiple and single waveforms, respectively. This process was repeated 4000 times, setting the filter properties (order, ripples, -3dB point) and the amplitude of multiplicative and additive noise using Monte-Carlo methods. Figure 1 shows an example of one waveform type including noise.

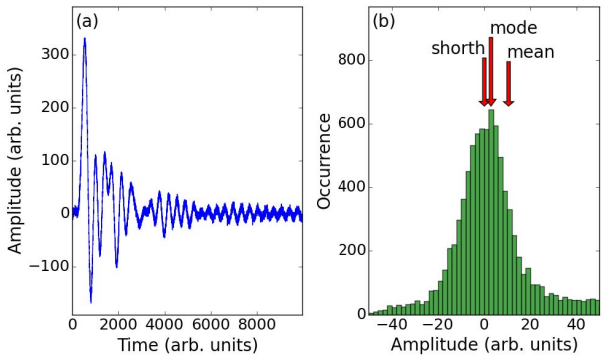


Fig. 1: (a) One of the 4000 different waveform types, which were analyzed, including noise. (b) Histogram of the waveform shown in (a). The histogram mode, the mean and the shorth derived from the waveform data are indicated by arrows.

III. Best Estimate and Uncertainty Calculations

The formulas shown here are taken from [3], as are the descriptions for the derivations of the formulas and their variables.

A. Mode of single waveform

The single waveform provides one mode value that gives *level*(*s*). The histogram that yielded *level*(*s*) comprised 500 bins ($N_{bin,0} = 500$). The combined uncertainty is given by

$$u_{mode,s} = \sqrt{\sigma_n^2 + u_{\text{bin}}^2 + u_0^2}\,, \qquad (1)$$

where σ_n is the standard deviation of the waveform values that are expected to represent noise. The second uncertainty component of (1) accounts for the finite bin width A_{bin} and is

U.S. Government work not protected by U.S. copyright

Bieler, Mark; Paulter Jr., Nicholas.
"Estimation of waveform state levels and uncertainties using the histogram and shorth methods."
Paper presented at Conference on Precision Electromagnetic Measurements (CPEM)2016, Ottawa, Canada. July 10, 2016 - July 15, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

expressed as $u_{bin} = A_{bin}/\sqrt{12}$. The last uncertainty component expresses the variation of the mode versus N_{bin}. To derive this component we varied N_{bin} between $N_{bin,0}/2$ and $2N_{bin,0}$ and calculated the deviation of *level*(s) with respect to the corresponding N_{bin}.

B. Mode of multiple waveforms

When N_w waveforms are available, the best estimate of *level*(s) is obtained from the mean of the N_w mode values and the uncertainty from

$$u_{mode,m} = \sqrt{u_{std,mode}^2 + u_{\mathrm{bin}}^2 + u_0^2}\,, \qquad (2)$$

with u_{bin} and u_0 being calculated as detailed in subsection *A* using the averaged waveform. The component $u_{std,mode}$ denotes the standard deviation of the mean of the N_w mode values.

C. Shorth of single *waveform*

The single waveform will provide a single shorth value taken as *level*(s) that is calculated with a shorth fraction $f_{s,0} = 0.5$ [4]. The uncertainty is obtained from

$$u_{shorth,s} = \sqrt{\sigma_n^2 + u_{L_{shorth}}^2}\,, \qquad (3)$$

where σ_n is identical to the definition in subsection *A*. The second uncertainty component of (3) accounts for the variation of the state level for different shorth fractions. To derive this component we have varied f_s between 0.3 and 0.7 and calculated the deviation of *level*(s) with respect to the corresponding f_s.

D. Shorth of multiple waveforms

When N_w waveforms are available, we calculate *level*(s) from the shorth of the averaged waveform, **Y,** and the uncertainty from:

$$u_{shorth,m} = \sqrt{u_{cov}^2 + u_{L_{shorth}}^2}\,. \qquad (4)$$

The first uncertainty component is given by $u_{cov} = \mathbf{H}_L\,\mathbf{\Sigma}_{\mathbf{Y}}\mathbf{H}_L^{\mathrm{T}}$, with $\mathbf{\Sigma}_{\mathbf{Y}}$ being the covariance matrix of the N_w waveforms. $\mathbf{H}_L$ is a row vector with the same length (N) as **Y**. Its elements are equal to $1/N$ if they belong to the shorth, and equal to zero otherwise. The superscript "T" denotes the transpose. The second uncertainty component is calculated as described in subsection *C* using **Y**.

IV. Discussion and Conclusions

The results of our simulations are shown in Fig. 2. The value of *level*(s) and its uncertainty, $u_{level(s)}$, can both be computed using the mode and shorth method to yield results that are realistic. For the waveforms studied here, the shorth method provides more accurate results than does the mode method. Yet, the shorth method for multiple waveforms is computationally expensive, such that the calculation of *level*(s) and $u_{level(s)}$ using the mode might be advantageous in certain situations. The data as shown in Fig. 2 do not distinguish the differences between the mode and shorth results depending on the type of waveform and type and magnitude of noise. Additional calculation methods along with different waveforms will be presented at the conference.

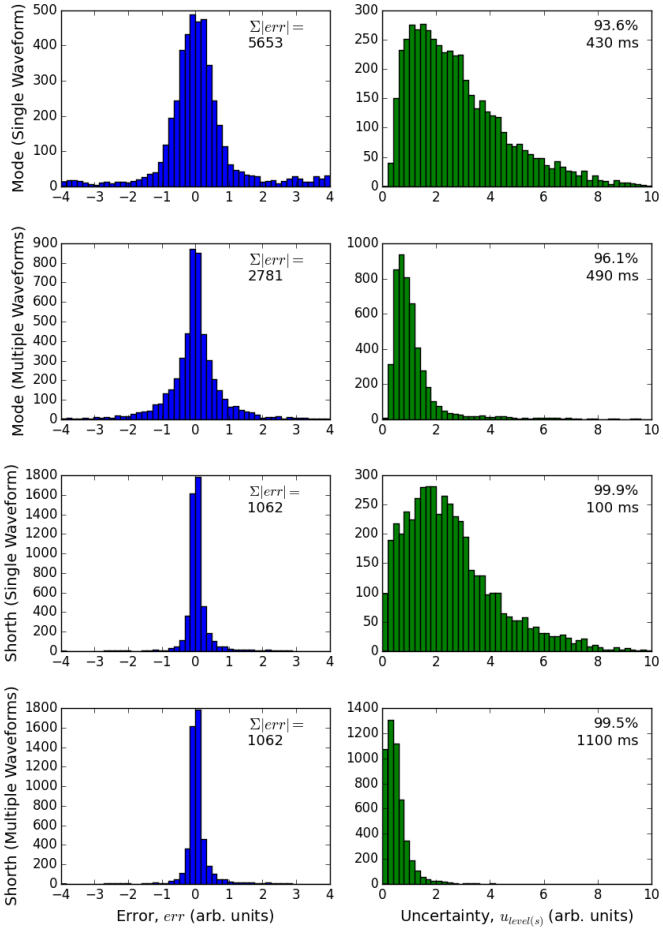


Fig. 2: Results of four different state level calculation methods for 4000 waveform types. Plotted are histograms of the errors in their *level*(s) and their uncertainties per equations (1) to (4). The sum over the absolute errors, $|err|$, is indicated in the upper right corner in the plots in the left column. The percentage shown in the upper right corner of the plots in the right column denotes how often the expanded uncertainty (k = 2) exceeds $|err|$ for the 4000 waveform types. The calculation time of each method is also indicated in the right column.

Acknowledgement

M.B. acknowledges support by the European Metrology Research Programme (EMRP). The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union.

References

[1] International Electrotechnical Commission, IEC 60469, Edition 1.0 2013-04, Transitions, pulses and related waveforms – Terms, definitions and algorithms.

[2] Institute of Electrical and Electronic Engineers, IEEE Std. 181-2011, IEEE Standard for Transitions, Pulses, and Related Waveforms.

[3] International Electrotechnical Commission, IEC 62754, "Computation of Waveform Parameter Uncertainties," in process.

[4] P.D. Hale and C.M.J. Wang, "Calculation of pulse parameters and propagation of uncertainty", IEEE Trans. Instrum. Meas. 58, 639 (2009).

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Bottom Up Approaches to Improved Polyolefin Measurements

Sara V. Orski, Thomas W. Rosch, Anthony P. Kotula, Richard J. Sheridan, Frederick R. Phelan Jr., Kalman B. Migler, Chad R. Snyder, Fernando Vargas Lara, Jack F. Douglas, and Kathryn L. Beers

National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA

Abstract

As a class of materials, polyolefins remain the largest production volume polymer in the world, as well as a highly desirable medium from which to engineer high performance and advanced properties for new applications. After decades of research, there are still major fundamental challenges to understanding the relationships between molecular structure, processing, morphology, and (ultimately) performance of these polymers. In two areas – surface adsorption for separation methods and melt crystallization – new tools and major advances in metrology are fueling a renewal of basic research into polyolefin characterization. Several approaches to these two problems, from theory and calculation to experimental models to elucidate each of these critical physical phenomena are undergoing current study at NIST.

Introduction

A major challenge in the measurement of polyolefins is being able to understand and quantify polyolefin behavior at the onset of a process. Very small changes in initial polymer orientation and interaction energy with the environment can shape the final state of the material. Accurate determination of the kinetics and thermodynamics of these changes is therefore critical to fully understanding polymer structure/property relationships. This knowledge gap is evident in the study of early polymer adsorption and crystallization processes. Polymer adsorption onto a surface is moderated by the equilibrium between polymer solvation near the surface and surface interaction energy. Crystallization of polyolefins is dictated by the nucleating structure in crystallization kinetics, which are often difficult to measure. Therein, our studies have approached the early polymer solvation and crystallization processes from two different angles – computational models that weigh included information for efficient simulation, and experimental models that use sensitive spectroscopic measurements to measure chain interactions. We will present current experimental and computational approaches at NIST to address polymer crystallization and solvation from first principles and elucidate these processes from the bottom up.

Surface Adsorption for Polymer Separations

The onset of polyolefin adsorption at surfaces requires thorough knowledge of the thermodynamics of the polymer/solvent/surface interface and has important applications in composite compatibility and in optimizing methods for polymer separations, where the interacting surface is often a small particle. These are challenging measurements, as the particle surface is often rough and non-spherical, and the polymer near the surface is highly concentrated and not in its native random coil configuration. Difficulties in heterogeneous surfaces and concentrated polymers are each addressed through novel approaches to computational and experimental models, respectively.

Progress in understanding, quantifying and predicting properties of complex shaped particles, polymers, and composites at NIST has been achieved through a decade of combination of theoretical, computational and experimental efforts. In particular, the development of algorithms and computational programs such as ZENO and advances in the DiMarzio-Rubin transfer matrix formalism[i] can provide precise calculations of polymeric-based material properties faster and in greater detail than prior computational tools. ZENO is a path-integral calculator that allows for precise computations of intrinsic and structural properties of polymer and particles having complex shapes.[ii,iii] These calculations can then be used to determine classical polymer composite properties, which are influenced by the particle surface interaction, particle shape, and property contrasts between particle and polymer matrix. The transfer matrix formalism accounts for local structure and enumerates the possible states of the material represented within a discrete lattice-type model. Each lattice site is weighted within the fixed probability matrix to adjust interaction energy and calculate thermodynamic properties of a confined polymer. This method permits a more realistic representation of the polymer/surface interface as complexity in the surface and polymer can be added to the model; this is evident in recent expansions of matrix theory that include additional lattice dimensions and

Beers, Kathryn; Douglas, Jack; Kotula, Anthony; Migler, Kalman; Orski, Sara; Phelan Jr., Frederick; Rosch, Thomas; Sheridan, Richard; Snyder, Chad; Vargas Lara, Luis Fernando.
"Bottom Up Approaches to Improved Polyolefin Measurements."
Paper presented at SPE Polyolefins Conference 2016, Houston, TX. February 21, 2016 - February 24, 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

heterogeneous boundary shapes and interaction energies.[iv] Specifically, calculations of surface lattice sites with heterogeneous interaction energies show a decrease in the phase transition temperature (critical condition) of an adsorbing polymer chain. This finding signifies that chemical heterogeneity at the surface requires greater binding strength for a polymer chain to adsorb. Recent method developments at NIST have focused on including molecular subunits of each monomer to account for non-linear and rigid chemical structures.[v] Models of individual monomers are built by forcing sequential steps in specific directions along lattice paths. A semi-flexible chain structure can also be studied by a "sub-matrix" approach that carries the history of the previous rigid monomer's direction and employs energetic barriers for specific future directions. As a result, this enables the analysis of the impact of more realistic structure and composition on critical conditions. Ultimately, matrices that can incorporate the most realistic picture of the adsorption will produce the most accurate partition coefficient between adsorbed and free polymer chains, resulting in better prediction of chromatographic separation modes such as liquid-adsorption, size-exclusion, and critical-condition chromatography.

Solvent quality of polymer chains in proximity to a surface also mediates affinity of the polymer to adsorb to a surface or remain in solution. We have developed an experimental model to approximate the near surface condition of an adsorbing chain and measure differences in solvation thermodynamics at high polymer volume fractions. The model uses vapor phase swelling of end-tethered polymer chains, or brushes, swollen by different organic vapors at known concentrations. X-ray reflectivity measurements of swollen polymer film thicknesses and scattering length densities were used to calculate the polymer-solvent interaction parameter of the confined film. As a proof of concept, this system was first studied using poly(methyl methacrylate) (PMMA) with organic solvents to study confined solvation at ambient temperature. The x-ray reflectivity curves for the PMMA brushes vapor swollen with organic solvents are shown in Figure 1a; the increase in the number of fringes is indicative of the thickness increase of the film. An initial probe of the PMMA brush with different saturated solvent vapors revealed that solvation thermodynamics of the brush does not follow the Flory solution model (lines in Figure 1b), as solvents with a similar solution Flory-Huggins interaction parameter (χ) gave large changes in the thickness swelling ratio α, relative to the dry state (Figure 1b).[vi] Consideration of the differences in molar volume of the organic solvents did not improve agreement with theory.

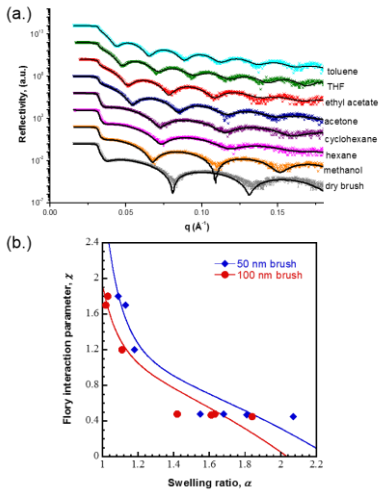


Figure 1. X-ray reflectivity curves of poly(methyl methacrylate) brush swollen with various organic solvents. (b.) Comparison of literature χ parameter with thickness swelling ratio α (swollen/dry) for a representative 50 nm and 100 nm brush trial, showing poor agreement of data with existing theory.[vi]

The experiment was then repeated by using acetone and systematically varying the vapor concentration to probe χ over a range of activities from the dry state to saturation (activity of 1). The χ values of the brush are greater than solution value ($\chi_{brush} = 0.6 \pm 0.1$; $\chi_{sol} = 0.46$) [‡] and decrease with increasing activity, indicating that the solvent quality is concentration dependent and improves closer to brush saturation. We hypothesize the observed decrease in solvent quality (i.e. higher χ) of the concentrated polymer at the surface, relative to dilute solution, will help drive surface adsorption, especially at high volume fractions of polymer. Concentration dependence of χ is especially critical when considering separation processes for polyolefin copolymers such as CRYSTAF and TREF. We are currently adopting the experiment to accommodate high temperature dissolution of polyolefins and to study solvation of linear and branched chains near structured carbon surfaces.

Advances in Quantifying Melt Crystallization

Nucleation and early stage crystallization dictate the final structure, processability, and mechanical properties of a semi-crystalline polyolefin. Much is still unknown about how polymer crystals form for both super-cooled and isothermal crystallization processes. NIST is studying the onset of both crystallization types through simulations and experiments, respectively.

[‡] Represented uncertainty is one standard deviation of the data.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Early stage nucleation of crystals from supercooled quenching of the melt are on the scale of a few nanometers in size with induction periods on a nanosecond scale. This brief window makes analytical measurement of these phenomena impossible with current technology, making simulation an ideal tool to probe crystallization kinetics. On the nanometer scale, the united atom model (UA) can accommodate the details of both bonded and non-bonded interactions, but cannot be extended to the micron-scale or industrially relevant systems using a reasonable amount of computational resources. New work at NIST generates a novel coarse-grain (CG) model that incorporates two UA sites per one CG site. This approach allows for UA level detail to inform the CG model through parameterization of non-bonded interactions with multiscale CG and bond length and angle information through iterative Boltzmann inversion, respectively. This model can address longer hydrocarbons and larger chain sampling to model interactions more relevant to polyethylene than short chain aliphatic compounds, demonstrated here by using a system of 60 chains of *n*-pentacontahectane (C 150) melted at 580 K (306.85 °C) and quenched to 280 K (6.85 °C). Results of the CG system are highly quantitative with UA simulations at equilibrium conditions of crystallization for both nucleation size and for the elongated structure of individual chains (Figure 2). Due to the absence of an induction period in the CG model, which is usually present in UA, the melting temperature and crystallization rate of *n*-pentaconahectane is higher for CG than for UA model, although the trends of both models are qualitatively consistent.

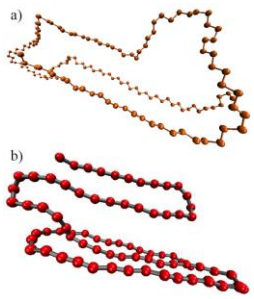


Figure 2. A single polyethylene chain after crystallization. Graph (a) corresponds to the united atom (UA) system and graph (b) corresponds to the coarse-grain (CG) system.[vii]

Complementary experimental work on isothermal crystallization of polyolefins has used Raman spectroscopy to distinguish local conformational states from orthorhombic crystals at a concentration of a few weight percent. Raman can determine relative concentrations of alkyl stretching, twisting, and bending modes that can be utilized to determine interchain and intrachain structures of the material. The transition mechanism of polyolefins from molten to semi-crystalline phase is highly debated and information about pre-crystalline structure can provide useful information about early polymer crystallization kinetics. Elongated chain conformations that are not part of the orthorhombic crystal, known as non-crystalline consecutive trans (NCCT) segments are of particular interest, as their concentration and distribution within the melt can enhance knowledge of chain dynamics during this induction period.

Crystallization of a linear polyethylene (NIST SRM 1475, nominal weight average molecular mass, (M_w) 53,000 g/mol) from the melt was measured by pressing a disk of the standard in a Linkam shear cell to a thickness 700 µm at 155 °C. Backscattered laser light from the sample is collected for Raman measurements, and transmitted light is collected to calculate turbidity and light depolarization. Measurements are performed during an isothermal crystallization wherein the sample is held at 130 °C after cooling from the melt temperature. Raman bands indicative of various modes of interaction for the melt and semi-crystalline state are shown in Figure 3a. Raman intensity for all spectra were scaled by the total integrated intensity in the CH_2 twisting region. The spectra were analyzed using the method described by Migler *et al.*, using normalized intensities of peaks representing the mass fractions of amorphous phase regions (1304 cm^{-1}), consecutive trans chain segments (1296 cm^{-1}), and crystalline phase segments (1415 cm^{-1}).[viii] This measurement assumes a three state model; the polyethylene can only exist as amorphous, crystalline or NCCT chain segments. The resulting mass fractions of all segments determined from Raman data are shown as a function of time after quench in Figure 3b. Some NCCT segments are present at the start of quenching in the predominately amorphous PE. Additional NCCT clusters form in conjunction with orthorhombic crystals at the beginning of polyethylene crystallization.

Measurement of turbidity and depolarization transmission taken simultaneously with the Raman data suggest crystallization measurement sensitivity is greater than Raman as crystalline regions were detected at mass fractions of approximately 0.01. Turbidity measurements indicate clusters forming pre-crystallization that are not homogeneously distributed through the material. Localized alignment of chains is also observed simultaneously from depolarized transmission measurements. This observation is due to nucleating of NCCT pre-crystalline domains from the melt state. This phenomenon can potentially add value in ultimately determining the kinetic route of crystallization and determining the structure of nucleation crystals.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

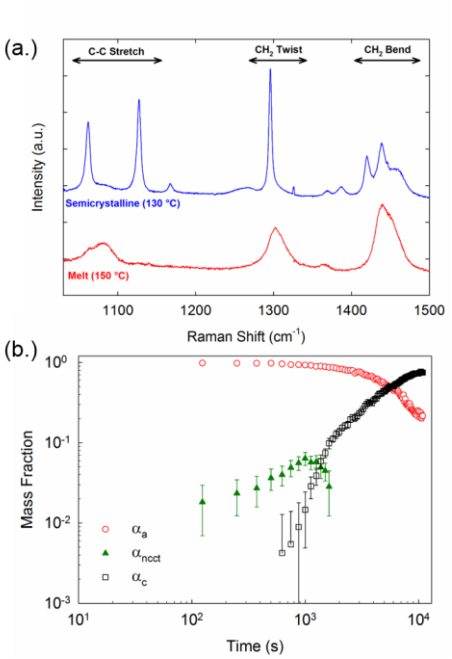


Figure 3. (a.) Typical Raman spectra of high-density polyethylene in the semi-crystalline state (T = 130 °C) and melt phase (T = 150 °C). The curves are scaled by the integrated intensity of the CH_2 twist region. (b) Evolution of the mass fraction of chain segments in the orthorhombic crystalline, the amorphous, and the non-crystalline consecutive trans (NCCT) states for HDPE following the 155°C to 128 °C temperature quench.[viii]

Conclusions

Sensitive scattering measurements such as Raman and x-ray reflectivity can be used to determine small compositional changes in polymer melts and thin films, as demonstrated in the measurements of low concentrations of pre-crystalline NCCTs in a polyolefin melt and organic solvent in a confined polymer thin film. Theoretical calculations can provide useful information about chain interactions on a finer scale than analytical instrumentation is capable of measuring. The use of theoretical and experimental models to measure subtle changes in polymer structure and composition will continue to be mutually beneficial to improve methodologies. This synergistic approach will be necessary in fundamental polyolefin research moving forward, as subtle changes in heterogeneous nucleation and complex branching architectures will affect multiple material properties. A systematic approach is required to improve materials design by understanding the intricacies of structure-property relationships and efficiently tailor new plastics for desired end-use properties. At NIST, it will be critical to inform design of the 21st century measurement technologies and reference materials necessary to support application of these materials.

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States.

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

[i] Di Marzio, E. A.; Guttman, C. M. e-print arXiv:1110.0142 [physics.gen-Ph] **(2011)**

[ii] http://web.stevens.edu/zeno/references.html

[iii] Mansfield, M. L.; Douglas, J. F.; Garboczi. E. J. *Phys. Rev. E* **(2001)**, 64, 061601

[iv] Snyder, C.R.; Guttman, C.M.; DiMarzio, E.A. *J. Chem. Phys.* **(2014)**, 140, 034905

[v] Guttman, C.M.; Synder, C.R.; DiMarzio, E.A. *Macromolecules* **(2015)**,48, 863-870

[vi] Figures reproduced from Orski, S.; Sheridan, R.; Chan, E.; Beers, K. *Polymer* **(2015)**, 72, 471-478 with permission of publisher.

[vii] Rosch, T.W.; Phelan, F. R, Jr. "Homogeneous Crystal Nucleation of a Coarse Grained Model of Polyethylene*"* *Macromolecules*. **Submitted October 2015**

[viii] Figures reproduced from Migler, K.B.; Kotula, A. P.; Hight Walker, A. R. *Macromolecules*, **2015**, 48 (13), pp 4555–4561 with permission of publisher.

Building Simulation Tools for Next Generation Soft Body Armour Testing Standards

S.P. Mates[1], A.L. Forster[1], M. Riley[1], K. Rice[1], and J. Ivancik[2]

[1]*National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, Md., USA, 20899-8553*
[2]*US Army Research Laboratory, 4502 Darlington Rd., Aberdeen Proving Ground, 21005, Md., USA*
smates@nist.gov

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Abstract. Accurate simulation tools for soft body armour testing would vastly improve a test designer's ability to optimize and refine standards by supplying quantitative information about the influence of test variables on test outcomes. Such tools would also facilitate the development of future armour testing standards that will leverage the improved understanding of impact injury mechanics by providing meaningful comparisons between actual human injury conditions and ballistic testing conditions. These potential benefits are real and well recognized. So, too, is the difficulty of developing such tools, which involve material systems ranging from the metal projectile to the woven polymer composite armour to the synthetic clay body surrogate. However, the emergence of the Digital Image Correlation (DIC) experimental method has greatly enhanced our ability to understand dynamic material behavior, making this difficult problem more tractable. This paper outlines our current vision for developing finite element simulation tools for soft body armour testing, including the types of measurement data and validation techniques being used, and describes preliminary simulation results of an idealized ballistic impact involving a rigid projectile striking clay-backed steel armour.

1. INTRODUCTION

The current National Institute of Justice (NIJ) Standard 0101.06 [1] for soft body armour testing continues to provide assurance to law enforcement departments across the United States that the body armour they acquire will protect officers on duty against the threats they were designed to defeat. An opportunity exists to provide even better assurances by leveraging considerable recent efforts to understand blunt impact injury. These efforts include the development of accurate, instrumented physical models designed to mimic human anatomy [2] as well as post-mortem human subjects (PMHS) [3] used to measure response to full-scale blunt impact and blast waves, detailed numerical models to investigate material-level effects and possible injury mechanics [4,5], and methods to quantify human injury using battlefield data [6]. Combined, these different elements promise to help us better understand and ultimately predict human injury under impact loads in a way that will allow engineers to better optimize body armour designs for law enforcement officers and soldiers alike.

Body armour testing standards can potentially leverage this new knowledge to provide more detailed assessments of armour protection level against ballistic impact than are afforded currently. A first step in this direction is to develop simulation tools that can reliably predict the ballistic standard test in its current form, so that the physics of current armour testing can be better quantified and compared to injury metrics and models developed under real field conditions. Accurate simulation tools are not easily developed due to the complex interplay between the wide variety of materials and interfaces involved in the ballistic impact problem. True predictive capability will likely remain elusive in the near future, given the complexity level and resource limitations. Mindful of this, in the present paper we outline our approach to developing simulation tools to better understand ballistic testing. In our favor is the fact that impact conditions are more tightly controlled compared to real field conditions. Projectile velocity is measured accurately, impact angle and mechanical constraints on the armour are controlled, and the backing material is pre-conditioned to provide a consistent response. Thus many sources of potential variability are already eliminated, improving chances that simulation tools might provide useful quantitative information.

Finally, in addition to the potential of providing insight into how current ballistic testing might compare to actual human injury response, simulation tools will allow test designers to investigate ways to improve body armour testing. Simulation tools will allow them to explore many important, practical test performance issues, such as the effects of inter-shot spacing, thermal or strength changes within the clay, bullet characteristics, and so on. Simulation tools could also help test designers transition testing to new clay formulations, other deformable backing materials, or to instrumented witness devices developed in the future. This paper outlines our current vision for developing finite element

Forster, Amanda; Ivancik, Juliana; Mates, Steven; Rice, Kirk; Riley, Michael.
"Building Simulation Tools for Next Generation Soft Body Armour Testing Standards."
Paper presented at Personal Armor Systems Symposium 2016, Amsterdam, Netherlands. September 19, 2016 - September 23, 2016.

simulation tools for soft body armour testing, describes the kinds of measurement data and validation techniques that are being incorporated into the model, and provides preliminary simulations of idealized ballistic impacts involving a rigid projectile striking a clay-backed steel armour using commercial finite element software.

2. MODEL ELEMENTS

2.1 Ballistic Clay

Ballistic clay has been investigated in compression over a range of temperatures and strain rates to determine its plastic response in our laboratory. Results have been fit to a Johnson-Cook viscoplastic material model developed originally for metals [7], given by Equation 1. The model was then compared to experimental results from the NIJ ball drop test [1] and the Prather impact test [8]. In the former, high-speed Digital Image Correlation (DIC) was used to track the ball impact on a standard clay box fixture (square box with inside dimensions measuring 610 mm on a side and 140 mm deep). In this test, a steel sphere measuring 6.35 cm in diameter impacts the clay box at a velocity of 6.26 m/s and with a kinetic energy of 20.44 J. The experiments were conducted at an average clay block temperature of 37 °C. An axisymmetric finite element model was constructed of the ball and clay block, which in the simulation measures 140 mm deep and has a radius of 305 mm, as shown in Figure 1(a). The block position was fixed along the side but not on the back surface, to mimic the experimental conditions. The ball was modeled as a rigid object, and the clay was modeled using the Johnson-Cook parameters found from the compression testing study, which are given in Table 1. The elastic response of the clay was linear with a Young's modulus of 2 GPa based on confined compression test results [9].

Results showed that while the final indentation depth was captured correctly, the model did not capture the elastic rebound of the clay, which was almost 25 % of the final indentation depth, as shown in Figure 1b. We concluded that a viscoelastic treatment of the clay elasticity was needed to capture the observed behavior, but that it was beyond the capability of the finite element software at hand. Regardless, the ball drop test demonstrated that the plastic response of the clay during this blunt, dynamic indentation test was captured well using the model derived from compression tests. The results are described in our previous PASS publication [9].

$$\sigma = \left(A + B\varepsilon^{n}\right)\left(1 + c\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)\left(1 - \left(\frac{T - T_0}{T_m - T_0}\right)^{m}\right) \tag{1}$$

Table 1. Johnson-Cook constants derived from quasi-static and dynamic compression tests.

Material	A [MPa]	B [MPa]	n	c[1]	m[2]
Roma Plastilina	0.00001	0.238	0.290	0.25	0.502

[1]Reference strain rate is 0.118 strain per second
[2]T_m = 100 °C, T_0 = 23 °C

Next, we used the model to predict a higher velocity impact test, called the Prather test [8]. In this test, a hemispherical projectile with a mass of 200 g is fired horizontally at a clay block that rests on a table. In the modeled test, the impact speed is 55 m/s, giving an impact kinetic energy of 315 J, much higher than the ball drop discussed earlier but only about half that of soft body armor ballistic tests, an idealized simulation of which is described next. The projectile penetration depth is determined in the experiment as a function of time using a high-speed camera. We note that the experiments were performed in an external laboratory (Experimental Facility 20, US Army Research Laboratory) with a different batch of Roma Plastilina #1 than the batch used in the experimental program at the National Institute of Standards and Technology (NIST) from which we derived our clay model. Figure 2 compares the axisymmetric model prediction of the projectile penetration history against the experimental results for an average block temperature of 37.4 °C ± 1 °C. The model captures the deceleration of the projectile qualitatively quite well, but shows a somewhat softer response (deeper penetration) compared with the experiment. The maximum penetration depth obtained from the simulation was approximately 1.2 cm deeper than the average value obtained from the experiments (9.0 cm ± 0.4 cm). Of note in the simulation is the importance of the friction coefficient between the impactor and the clay, which was taken to be rather high given the nature of the clay. Friction has a

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

significant effect on the penetration depth, and bears further attention going forward. We also note that the experiment showed no sign of elastic springback as was observed in the ball drop test. Despite the imperfect agreement with the experiment, the similarity was encouraging, given the fact that the clay used to derive the model was different in age and batch identification from the clay used in the Prather test.

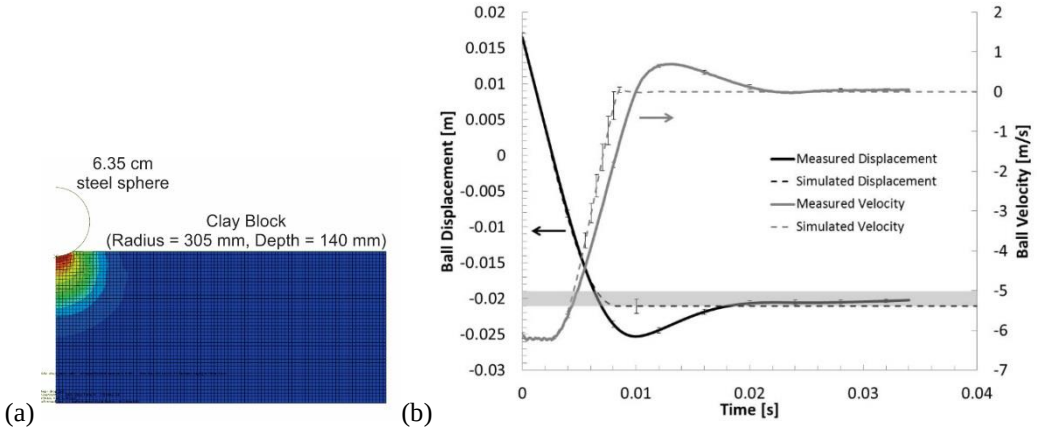


Figure 1. (a) Computation domain for ball drop test simulation. (b) Comparison of simulated and measured ball position and velocity history during the test.

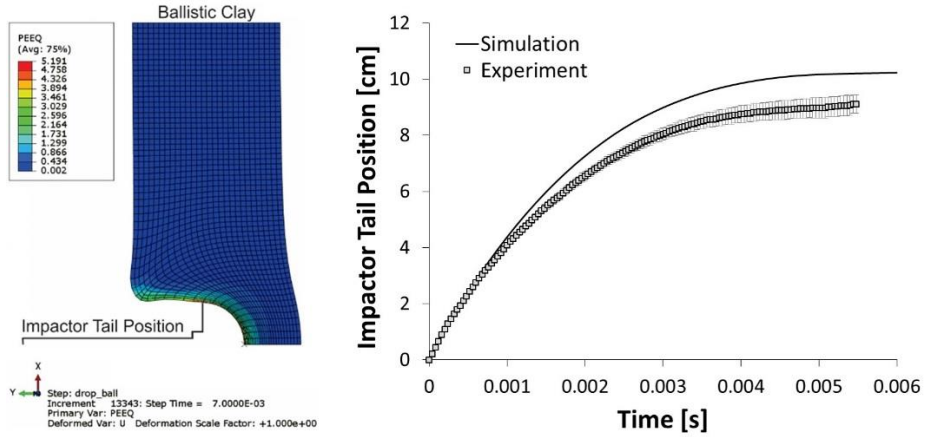


Figure 2. Final frame of a finite element simulation of Prather test (left) and a comparison of the simulated impactor tail position history with experimental results (right) for an average impact speed of 56.4 m/s ± 1 m/s and an average clay temperature of 37.4 °C ± 1 °C for a total of 19 experiments.

3.2 Deformable Projectiles

A potentially significant issue in ballistic testing is the deformation behavior of the test bullets. Experience shows that nominally similar bullets, in terms of mass and caliber, can perform differently in ballistic testing. While the most prominent projectile effects can be attributed to the sometimes large differences in the strength of the component materials (steel versus lead), there may exist more subtle differences among nominally similar bullets that can affect ballistic test results. We examined methods to develop accurate simulations of several different projectiles used in the current NIJ Standard. Model predictions of dynamic compression in whole bullets were explored using direct-impact Kolsky bar experiments, where the bullet is held fixed at the impact end of a Kolsky bar where it is struck by a massive striker moving at moderate velocity. In the example described in [10], a .40 caliber Smith and Wesson (S&W) Full Metal Jacket (FMJ) projectile is struck at 15.3 m/s with a kinetic energy of 62.5 J, which is about 10 % of the energy of an NIJ standard soft body armour shoot test for this bullet type

(reference velocity of 325 m/s, bullet weight of 11.66 g, kinetic energy of 617 J). The impact produces a total axial engineering strain of 0.35 in the bullet. The bullet deformation history was captured using three-dimensional, high-speed digital image correlation (3D DIC) [11], and the measurements were compared to an axisymmetric finite element model of the test.

A multitude of simulations were performed where design-of-experiments techniques were used to investigate the sensitivity of the material parameters and to find the best match between the simulation and the measured deformation and impact load history. Using in-house developed material parameters for the lead core, we found that the improved constitutive constants for the jacket material were much stronger than the literature values for cartridge brass, the material though to most closely mimic the copper alloy jacket for which constitutive data are available [12]. The comparison between the baseline and optimized simulations of the bullet deformation history and the experiments, repeated from [10], is shown in Figure 3. Overall, despite the smaller impact velocities and lower deformation levels present in the Kolsky bar experiment, this method helps improve the accuracy of whole bullet models for simulating ballistic testing. Full details on the .40 S&W and the .357 Jacketed Soft Point (JSP) projectile impact tests are given in [10] and [13], respectively.

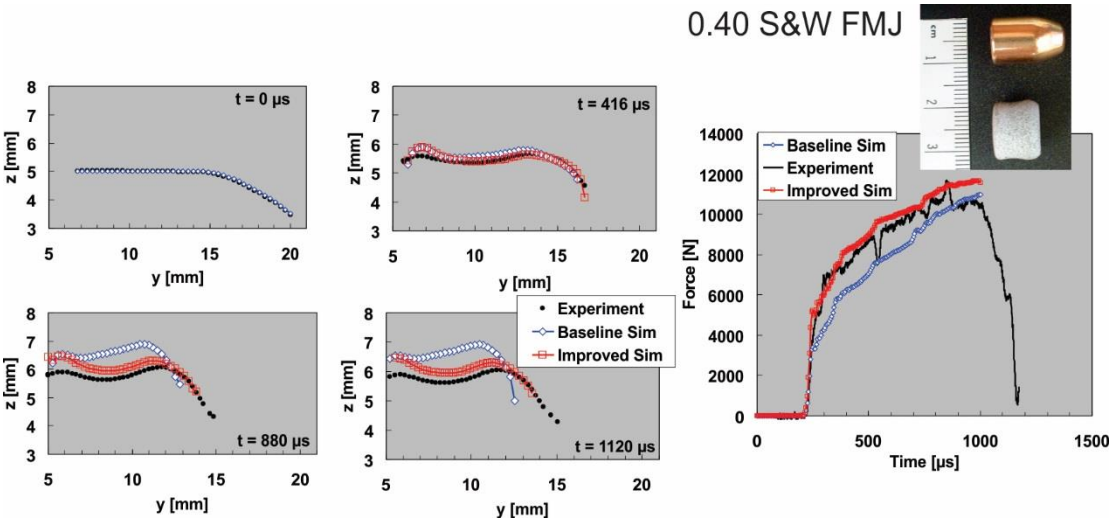


Figure 3. Comparison of baseline and improved simulation results derived from design-of-experiments methods that explore the effect of model parameter variations for the jacket material on the shape history and force history measurements.

3.3 Preliminary Ballistic Test Simulations

To explore the performance of the clay model under real ballistic test conditions, we simulated an idealized ballistic test using simplified armour and projectile models. The clay model was identical to the one used previously to model the ball drop test and the Prather impact test. For the armor, we used a simple steel plate placed in front of the clay block. The intent of this simulation was to observe the behavior of the clay model under real ballistic impact velocities and not, at this time, to explore the effect of armor or bullet behavior, which is obviously more complex. The armour plate was made of 4340 steel and was 5 mm thick and 300 mm in diameter. The steel has a Young's modulus of 200 GPa and Johnson-Cook (Equation 1) plasticity parameters of A = 792 MPa, B = 510 MPa, n = 0.26 and C = 0.015 obtained from [14]. We used a rigid, spherical projectile with a diameter of 20 mm impacting at 322 m/s and having a mass of 11.7 g, equivalent to a Type IIA armour test shot. The diameter of the rigid projectile was chosen to mimic the mushrooming impact behavior of a lead-cored projectile against soft body armour. At this time we have yet to incorporate our deformable bullet model into the ballistic test simulation, which requires special adaptive meshing techniques to handle the extreme deformations expected in the bullet.

The finite element solution is shown in Figure 4 at three times during the impact process, while Figure 5 plots the positions of the projectile, the rear surface of the armour, and the top surface of the clay on the axis of symmetry throughout the impact simulation. As the plots show, soon after impact the simulation predicts a separation between the clay surface and the back face of the armour, which grows in time to substantial proportions. The permanent indentation left in the clay was 30 mm deep,

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

whereas the maximum indentation in the armour plate was less than 7 mm. This separation phenomenon has been shown experimentally by high-speed X-ray photography in a previous PASS conference [15]. Their work also pointed out significant elastic rebound in the clay during ballistic impact, which was also observed in our own ball drop experiments using DIC measurements of the ball impact. Elastic rebound is not captured in the present model for reasons discussed earlier. Overall, the simulated plastic impact behavior of the clay seems consistent with experimental observation.

We are currently investigating how to better validate the simulations quantitatively, which remains challenging due to the extremely short timescales and large deformations involved. Examples of high-speed DIC measurements of armour panels during ballistic impact are prevalent in the literature [16], and we believe that this kind of experiment is a useful approach. We plan to report on progress in this area, as well as on incorporating deformable bullet models developed in-house and woven armour models from the literature, in future PASS conferences.

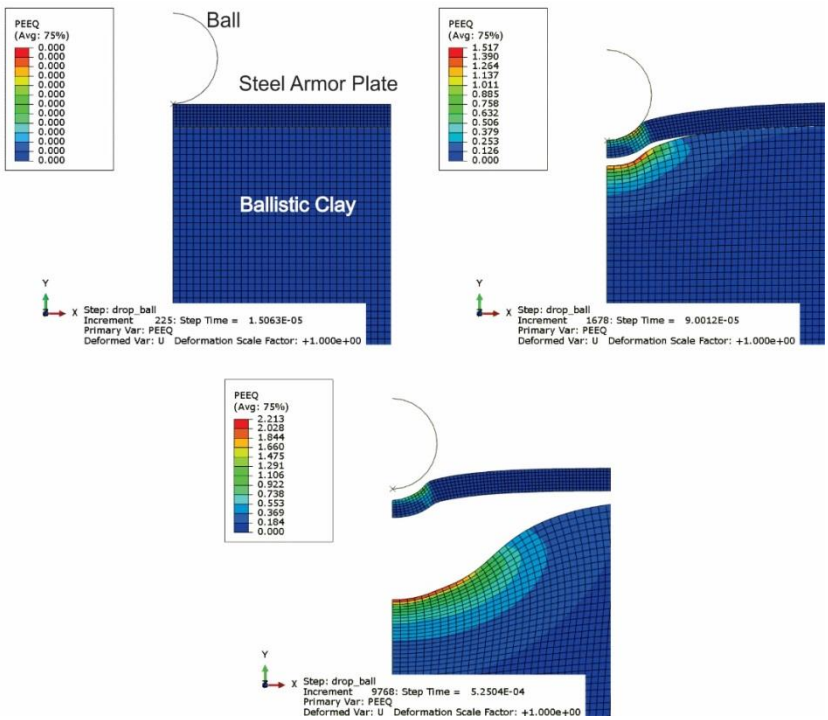


Figure 4. Simulation of a 20 mm rigid sphere impacting at 322 m/s on a 5 mm thick steel armour plate backed by a standard ballistic clay fixture at 37 °C. Results are shown at three time points: prior to impact (top left), just after armour-clay separation (top right), and well after armour-clay separation (bottom). Step time units are seconds. PEEQ stands for equivalent plastic strain.

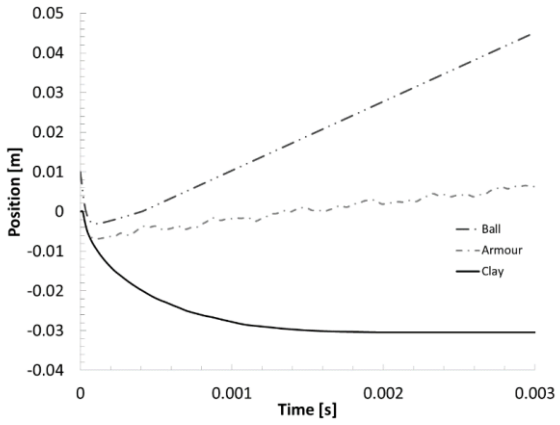


Figure 5. Position histories along the axis of symmetry of the impacting sphere, the back surface of the armour, and the top surface of the clay during a ballistic impact simulation at 322 m/s.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

4. CONCLUSIONS

We have outlined our vision of and progress toward creating finite element simulation tools for soft body armour testing. If successful, these tools can be employed by the testing community to improve and refine ballistic testing, making standard test methods even more robust and helping to draw more substantial links between testing conditions and actual blunt human injury.

Our current state of model development is as follows:

1) Material models for ballistic clay have been developed that show good quantitative agreement with the ball drop test and good qualitative agreement under actual ballistic impact conditions in terms of plasticity. Efforts are needed to better characterize the viscoelastic portion of the material response to fully capture the impact behavior of the clay.

2) A method for establishing models to predict dynamic deformation of the types of projectiles used in NIJ armour testing standards has been developed based on simulating dynamic, large strain axial impact tests on such bullets. Ballistic test simulations will be attempted using deformable bullet models run within commercial finite element codes that can be made to account for the extreme plastic strains that develop during ballistic impact, most likely via adaptive re-meshing techniques.

3) Accurate models of woven soft body armour are complex, and their continuing development remains at the forefront of ballistics research. As such, for the short term we will include idealized armour models that provide acceptable macro-scale response.

4) We are working to develop experiments that can provide time-resolved data during actual ballistic testing experiments. These data can be used to assess the accuracy of our ballistic test simulations, which bring together the data and model elements thus far developed that have been outlined in this paper.

Acknowledgements and Disclaimers

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] NIJ Standard-0101.06. Ballistic Resistance of Body Armour. Washington, DC (USA): U.S. Department of Justice, 2008.
[2] Biermann PJ, Ward EE, Cain RP, Carkhuff BG, Merkle AC and Roberts JC, J. Adv. Mater., 38 (2006), 3-12.
[3] Wilhelm M, and Bir C, Forensic Sci. Int., 174 (2008), 6-11.
[4] Roberts JC, Ward EE, Merkle AC and O'Connor JV, J Trauma, 62 (2007), 1127-1133.
[5] Shen X, Niu Y, Bykanova L, Laurence P and Link N, "Characterizing the Interaction Among Bullet, Body Armor, and Human and Surrogate Targets," J. Biomech, Eng.-T. ASME, 132 (2010).
[6] Champion HR, Holcomb JB, Lawnick MM, et al., J Trauma, 68 (2010), 1139-1150.
[7] Johnson GR and Cook WH, Proc. 7th Int. Symp. Ballistics, 1983.
[8] Prather RN, Swann CL and Hawkins CE, Army Technical Report ARCSL-TR-77055, 1977.
[9] Mates SP, Riley M, Forster A and Rice K, "Mechanical Behavior of Ballistic Clay as a Function of Temperature, Pressure and Strain Rate," Personal Armour Systems Symposium 2014, Cambridge, UK, Sept. 2014.
[10] Mates SP, Rhorer RR, "Dynamic Deformation of Copper-Jacketed Lead Bullets: Experiments and Modeling," Personal Armour Systems Symposium 2010, Quebec City, Canada, Sept. 2010.
[11] Sutton MA, Orteu JJ and Schreier HW, Image Correlation for Shape, Motion and Deformation Measurements, (Springer, New York, 2009).
[12] Johnson GR, Hoegfeldt JM, Lindholm US and Nagy A, ASME J. Eng. Mat. and Tech., 105 (1983) 42-47.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

[13] Mates SP, Rhorer RR, "Dynamic Deformation of Copper-Jacketed Lead Bullets Captured by High Speed Digital Image Correlation," Society for Experimental Mechanics 2010 Annual Meeting, Indianapolis, IN, June 7-10, 2010.
[14] Meyers MA, The Dynamic Behavior of Materials, (Wiley-Interscience, New York, 1994), p.328.
[15] Broos JPF, van der Jagt-Deutekom M, Halls VA, Zheng JQ, "Separation Between Armour and Clay Backing during Projectile Impact," Personal Armour Systems Symposium 2012, Nuremburg, Germany, Sept.17-21, 2012.
[16] O'Masta MR, Compton BG, Gamble EA, Zok FW, Deshpande VS and Wadley HNG, Int. J. Impact Eng., 86 (2015), 131-144.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

High-Strain-Rate Deformation of Ti-6Al-4V through Compression Kolsky Bar at High Temperatures

S. Gangireddy, S.P. Mates

Materials Science and Engineering Division, MML, NIST, Gaithersburg, MD, USA

ABSTRACT
In this paper, we present our first results from the study of the constitutive response of a popular Titanium alloy, Ti-6Al-4V, using a variation of the compression Kolsky Bar technique that employs electrical pulses to achieve high temperatures. Experiments are conducted at temperatures ranging from room temperature to 1000 °C at a strain rate of about 2200 s^{-1} and a heating rate of about 1500 °C/s. The dynamic stress-strain results demonstrate significant thermal softening in the alloy that could be described by Johnson-Cook equation with m = 0.8 up to 650 °C. Above 650 °C the rate of change in the flow stresses was faster, which is attributed to allotropic transformation that results in a change in the phase fractions of the hcp and bcc phases present in the alloy. Evidence of transformation is observed in the microstructure of post-compression specimens, which showed an acicular morphology formed from the high temperature bcc phase on quenching.

KEY WORDS: Kolsky Bar, Ti-6Al-4V, High-Temperature, Dynamic-Response, Thermal-Softening

INTRODUCTION

Methods for mechanical testing under rapidly applied loading are well established, such as the Split-Hopkinson Pressure Bar (SHPB) or Kolsky Bar technique. Variations of Kolsky Bar with additional components for achieving high temperatures exist as well. Typically miniature furnaces, induction coils, or radiation heating are used [1, 2, 3]. Heating times associated with these methods are typically on the order of minutes. At NIST, a new variation of Kolsky Bar system has been developed [4, 5] where electrical current is pulsed directly through the sample while it sits fixed between the incident and transmission bars. This technique gives us a unique advantage of rapid heating rates, up to 6000 K/s, where the heating time can be less than one second. Such rapid heating combined with rapid loading conditions creates a closer simulation to extreme physical processes such as high speed machining, explosive impact and other highly dynamic thermo-mechanical events. These results can give valuable insights that may lead to improved cutting processes or development of better blast-resistant structures.

Ti-6Al-4V is a widely used titanium alloy with aerospace applications owing to its excellent combination of high specific strength and corrosion resistance [6]. Pure titanium is allotropic and undergoes transformation from hcp (α) to bcc (β) crystal structure at elevated temperatures. The alloy Ti-6Al-4V contains both alpha stabilizing Aluminum (6 wt %) and beta stabilizing Vanadium (4 wt %) and consists of a mixture of α + β phases at room temperature [7]. At higher temperatures, the phase fraction of β increases, and above a temperature of about 995 °C the alloy is 100 % β [8]. This is known as the β-transus temperature. The mechanical response of this material can therefore be expected to be strongly dependent on temperature. The temperature sensitivity of the dynamic mechanical response of a commercial Ti-6Al-4V alloy was investigated using NIST's electrical pulse-heated Kolsky Bar system in the temperature range of 23 °C to 1008 °C under a strain rate of about 2200 s^{-1}.

EXPERIMENTAL PROCEDURE

A commercial Ti-6Al-4V alloy of composition, 6.65 % Al, 4 % V, 0.02 % C, 0.23 % Fe, 0.007 % N, 0.197 % O, 0.003 % H, is studied in this investigation. The as-received material's microstructure consisted of globular α phase grains in a β matrix. The alloy, purchased in the form of a 2 mm thick plate, was cut using electrical discharge machining (EDM) to obtain the compression samples in cylindrical form of 4 mm diameter and 2 mm thickness. These samples were placed in between the incident and transmission bars of 150 mm diameter. Electrical current is conducted directly through the sample using the bar ends as electrodes. Owing to the large difference in the sample and bar cross-sectional areas and the very short heating times involved, the sample alone heats up while the bars themselves remain cool, with a less than 25 °C temperature rise even

while the samples heated to 1000 °C [4]. In this series of experiments, the total heating time, which includes a transient heating period followed by a hold period, is limited to 3.5 seconds.

The temperature of the sample is monitored through three signals: a thermocouple spot welded onto the sample surface and two fast response infrared spot pyrometers focused on opposite sides of the sample surface. During heating by the electrical current, the thermocouple signal gets affected by electromagnetic interference. Hence one of the pyrometers is used as a feedback sensor to a specialized power supply that controls the sample temperature. The second pyrometer is used to monitor temperature uniformity. The pyrometer signal measures only the radiance temperature of the specimen. The thermodynamic, or true, temperature is determined from the thermocouple signal, which takes a few milliseconds to settle after the current is turned off. Hence compression pulse is timed to arrive about 20 ms after the current is switched off so a clean thermocouple signal can be obtained prior to impact. Just at the time of impact, the pyrometer signals could also be used to obtain radiance temperature, but after impact the specimen moves out of the view of pyrometers and the signals are lost. The radiance and true temperatures at impact, plotted in Figure 1, are linearly correlated, indicating that even using the pyrometer as the feedback control signal, good true temperature control is possible.

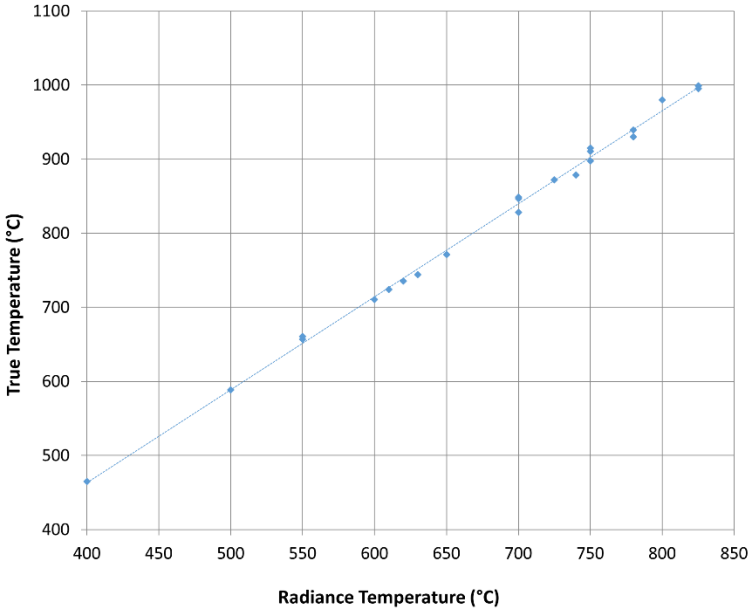


Fig. 1: Thermodynamic or true temperature measured by the thermocouple and the radiance temperature measured by the pyrometer at the time of impact show a strong correlation between 400 °C and 1000 °C.

The usual analysis method for deducing dynamic stress-strain from strain gauge signals is performed with a correction for the presence of graphite foils used for uniform contact conductance between the sample and the bars. The foil and the sample are treated as separately deforming elements and the foil response is deducted from the overall contraction between the compression bars to obtain the sample contraction. The detailed analysis for this deduction of dynamic stress-strain curves is presented elsewhere [4].

RESULTS AND DISCUSSION

The dynamic true stress- true strain curves from compression tests conducted at temperatures ranging from 23 °C to 1008 °C under a strain rate of 2200 s^{-1} (± 800 s^{-1}) are depicted in Figure 2. The total time of heating is 3.5 seconds before the sample is impacted. The test temperature quoted is the true temperature measured using the thermocouple at the moment the first compression pulse hits the sample.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

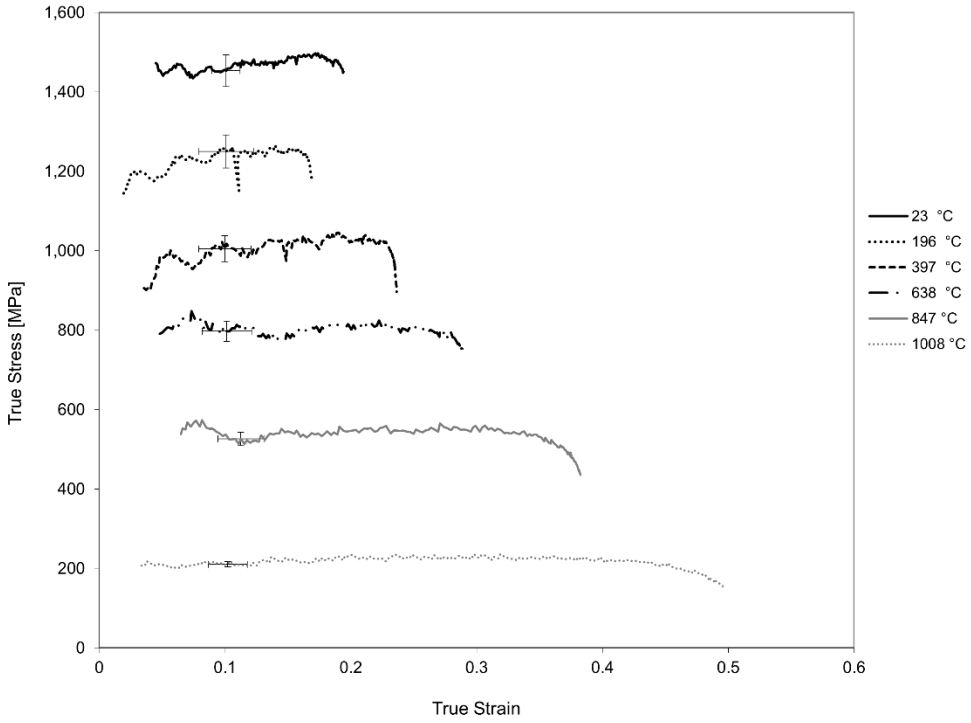


Fig. 2: Dynamic true stress – true strain graphs of Ti-6Al-4V alloy undergoing compression under a strain rate of 2200 s^{-1} (± 800 s^{-1}) at temperatures between 23 °C and 1008 °C.

Thermal softening is evident, as the flow stress decreases significantly with increase in temperatures. The flow stresses at 0.1 and 0.15 true strains are plotted as a function of test temperature in Figure 3 (a). The Johnson-Cook empirical equation is often used to describe material response as a function of plastic strain, strain rate and temperature [9]:

$$\sigma\left(\varepsilon_p,\ \dot{\varepsilon}_p, T\right) = \left[A + B\left(\varepsilon_p\right)^n\right]\ \left[1 + C\ ln\left(\ \dot{\varepsilon}_p{}^*\right)\right]\ [\ 1 - (T^*)^m\ ] \qquad (1)$$

where thermal softening is encompassed by the last term, $[\ 1 - (T^*)^m\ ]$. T* is the homologous temperature given by $(T-T_{ref})/(T_{melt}-T_{ref})$. In an attempt to describe our results similarly, the normalized flow stresses $\sigma(T)/\ \sigma(T_{ref})$ at 0.15 true strain are plotted as a function of T* with the reference temperature, T_{ref} = 23 °C, and melting point, T_{melt} = 1630 °C. This graph in Figure 3(b) shows that the data could be well described by the Johnson-Cook equation for temperatures below 650 °C with a thermal softening parameter *m* of value 0.8. Similar values of *m* have been reported in other high strain rate investigations such as: Seo et al [3] whose *m* was 0.7 while Johnson [10] reported *m* =0.8. Dorogoy et al [11] have reported a similar *m* of 0.8 from quasi-static loading conditions as well. At temperatures higher than 650 °C however, there is a steeper reduction in the flow stresses.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

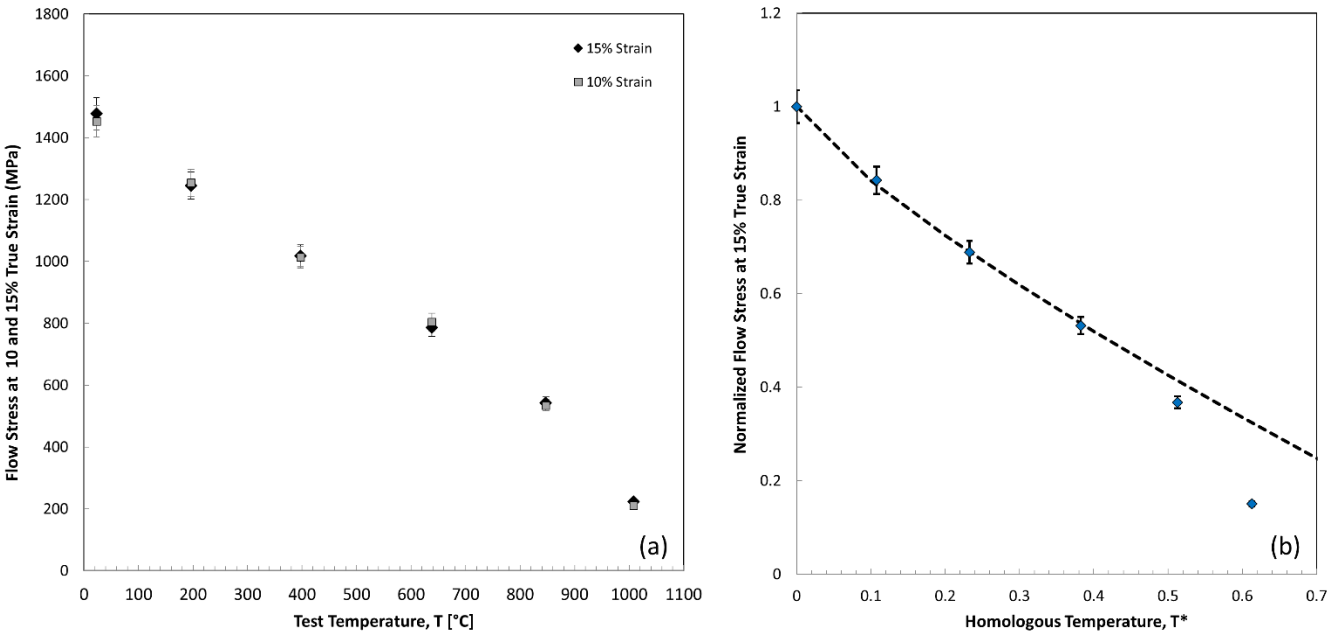


Fig. 3 (a) Flow stresses at 0.1 strain and 0.15 true strain plotted as a function of thermodynamic temperature. **(b)** Normalized flow stresses $\sigma(T)/\sigma(T_{ref})$ at 0.15 true strain plotted as a function of homologous temperature $T^* = (T-T_{ref})/(T_{melt}-T_{ref})$, with T_{ref} = 23 °C, T_{melt} = 1630 °C.

This behavior can be a result of the ongoing allotropic transformation in Ti-6Al-4V. At room temperature, pure Titanium exists in hcp (α) crystal structure and when heated to 882 °C, it changes into bcc (β) [7]. The alloy Ti-6Al-4V containing both α-stabilizing Aluminum and β-stabilizing Vanadium elements does not have a sharp transition point, instead there is a temperature range in which there is a gradual transformation. The start temperature α-transus is lower than room temperature, and the β-transus is about 995 °C [8]. At room temperature, the alloy contains both α and β. At higher temperatures, α starts transforming into β, increasing the β phase fraction. The increase in β volume fraction is insignificant below 600 °C, after which it starts to rise slowly initially and becoming rapid at elevated temperatures, especially closer to the β-transus. The α phase is known to show better high-temperature strength than the β phase [12], so at temperatures above 600 °C, it can be expected that the flow stresses show a more rapid decline with increasing temperature.

It has to be considered here that the heating time involved in our tests is only 3.5 seconds. Whether this time is sufficient for the α->β transformation needs to be verified before concluding that the low flow stresses observed are a result of the allotropic transformation. Confirmation of this transformation will be found in the microstructure of post-compression specimens. The electrical-pulse heated Kolsky Bar allows rapid cooling rates of about 1000 °C/s owing to the swift heat transfer from the hot specimen to the cool bar ends once the heating current is turned off. If the test temperature is higher than the martensitic start temperature M_S, this quenching causes any new β produced by the α->β transformation on heating to transform into martensitic acicular α′ [12].

Figure 4 compares the microstructure of the as-received material with the microstructure of a post-compression specimen, tested at 850 °C, both etched with Kroll's Reagent. Figure 4 (a) depicts the initial/ as-received microstructure consisting of globular α grains (bright contrast) in a matrix of β (dark contrast). At room temperature, the phase distributions consists of about 90 % α and 10 % β by volume. Figure 4 (b) shows a region in the post-compression specimen tested at 850 °C consisting of a large area of acicular α′. According to the phase diagram [12], the phase composition at 850 °C is expected to be 66 % α and 34 % β. When quenched from this condition, the volume fraction of primary α remains the same, but the β starts to transform into acicular α′. Since the martensite finish temperature M_F is lower than room temperature, the transformation is incomplete, leaving some β remaining in the microstructure. The quenched microstructure can therefore be expected to contain about 66 % primary α, 24 % α′ and 10 % β. Acicular α′ is easily distinguishable from the equiaxed primary α as can be seen in Figure 4(b), and hence its appearance confirms the allotropic transformation.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

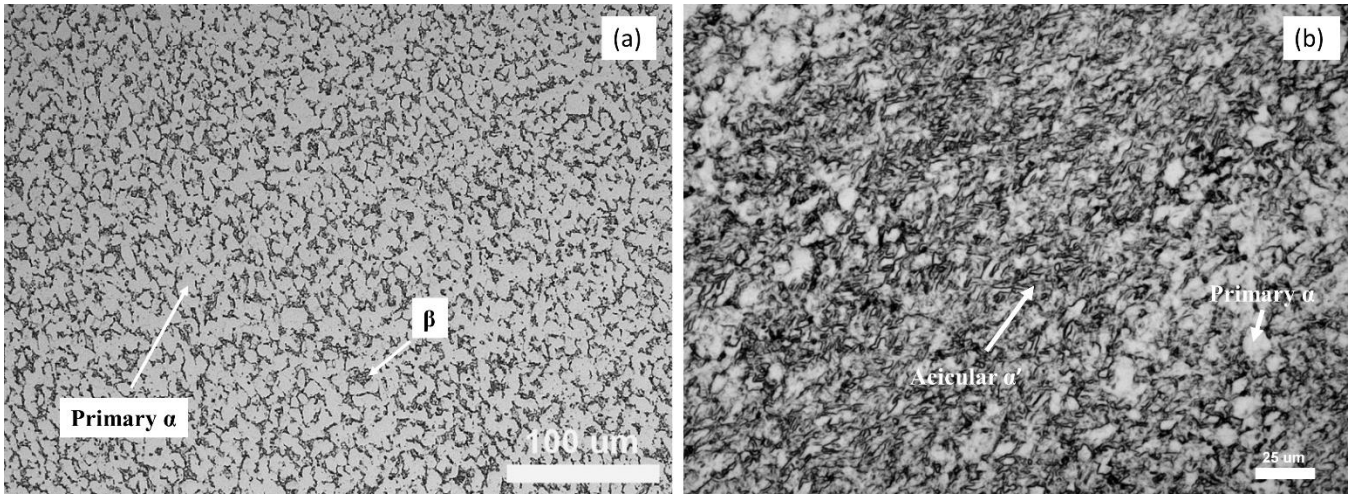


Fig. **4 (a)** Microstructure of as received Ti-6Al-4V **(b)** Microstructure of Ti-6Al-4V post-compression specimen tested at 850 °C.

The initial results indicate that the constitutive response of Ti-6Al-4V from the pulse heated Kolsky Bar system is comparable to the behavior reported by other investigators [1, 3, 14] from room temperature to about 650 °C. However, at higher temperatures we observe the thermal softening could not be similarly described using Johnson-Cook's equation owing to the allotropic transformation, which has not been considered by most of the other studies [1, 3, 14]. Modifications to the Johnson-Cook equation are often made to account for phase transformation [12], but they are essentially step functions only suitable for well-defined transformation temperatures. But in a material such as Ti-6Al-4V, where there is a continual phase transformation in a two-phase field, they are not ideal. A new modification to the Johnson-Cook equation is therefore essential to describe thermal softening in Ti-6Al-4V, one that considers the changing phase fractions together with the difference in the high temperature strengths. We propose to conduct more experiments in the 700 °C to 1000 °C range with smaller temperature steps to more completely describe Ti-6Al-4V's thermal softening behavior through the transition region.

CONCLUSIONS

A compression Kolsky Bar system, which directly pulses electrical current through samples for achieving high temperature, is used to study the widely used Titanium alloy Ti-6Al-4V. The first results from this investigation where compression tests were conducted at about 2200 s^{-1} strain rate at temperatures ranging from 23 °C to 1008 °C are reported in this paper. The results show that the dynamic response of the material has a strong dependence on temperature. The thermal softening occurs more rapidly above 650 °C, and is attributed to allotropic transformation from hcp (α) to bcc (β) crystal structure. Evidence of this transformation is discovered in the microstructures of post-compression specimens, which show an acicular α′ phase formed from quenching of β when the test temperatures are above martensite start, M_S. Further data is essential, especially in the 700 °C to 1000 °C temperature range, to develop a new modification to the Johnson-Cook equation, which considers the changing phase fractions as a function of temperature to be able to describe the Ti-6Al-4V's mechanical response.

DISCLAIMER

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

REFERENCES

[1] Lee WS, Lin CF (1998) Plastic deformation and fracture behavior of Ti-6Al-4V alloy loaded with high strain rate under various temperatures. Materials Science and Engineering A241:48-59.

[2] Seo S, Min O, Yang H (2005) Constitutive equation for Ti-6Al-4V at high temperatures using the SHPB technique. International Journal of Impact Engineering 31 (6): 735-54.
[3] Gray GT III (1990) Classic Split Hopkinson pressure bar testing, ASM Handbook 8: 462 – 476. Materials Park, Ohio.
[4] Mates S P, Rhorer R, Whitenton E, Burns TJ (2008) A pulse-heated Kolsky Bar Technique for Measuring the Flow Stress of Metals at High Loading and Heating rates. Experimental Mechanics 48:799-807.
[5] Basak D, Yoon HW, Rhorer R, Burns TJ, Matsumoto T (2004) Temperature control of pulse heated specimens in a kolsky bar apparatus using microsecond time resolved pyrometry. International Journal of Thermophysics 25 (2):561-574.
[6] Wood RA (1972) Titanium Alloy Handbook. Metals and Ceramics Information Center Battelle Publication No. MCIC-HB-02.
[7] Froes FH (2015) Titanium – Physical Metallurgy, Processing and Applications. ASM International.
[8] Semiatin SL, Seetharaman V, Weiss I (1997) The thermomechanical processing of Alpha/Beta Titanium Alloys. JOM 49:33-39.
[9] Johnson GR, Cook WH (1983) A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. Proceedings of the 7th International Symposium on Ballistics Vol. 21:541-547.
[10] Johnson GR (1985) Strength and fracture characteristics of a titanium alloy (0.6Al, 0.4V) subjected to various strains, strain rates, temperatures and pressures. Technical report TR 86-144, Naval Surface Warfare Center.
[11] Dorogoy A, Rittel D (2009) Determination of Johnson-Cook material parameters using the SCS specimen. Experimental Mechanics 49: 881.
[12] Donachie MJ (2000) Titanium: A Technical guide. ASM International.
[13] Andrade U, Meyers MA, Vecchio KS, Chokshi AH (1994). Dynamic recrystallization in high strain, high strain rate plastic deformation of copper. Acta Metallurgica et Materialia 42: 3183-95.
[14] Nemat-Nasser A, Guo WG, Nesterenko VF, Indrakanti SS, Gu YB (2001) Dynamic response of conventional and hot isostatically pressed Ti-6Al-4V alloys: experiments and modeling. Mechanics of Materials 33 (8): 425-39.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Opportunities for Inverse Analysis in Dynamic Tensile Testing

Steven Mates[1], Fadi Abu-Farha[2]

[1]National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD, USA
[2]Clemson University - International Center for Automotive Research, Greenville, SC 29607, USA

ABSTRACT
Dynamic tensile testing using Kolsky Bar methods are used to assess crashworthiness of new structural materials needed for lightweight automotive design. High speed Digital Image Correlation (DIC) measurements routinely show that the strain experienced by dynamic tensile specimens deviates markedly from what is expected from the original strain wave analysis techniques used in tensile Kolsky bar metrology. Deviations can be manifest either by different average strain values over the gage section, or by departures from strain uniformity, or both. The former can be attributed to plastic yielding in the specimen outside the gage section, while the latter concerns specimen geometry and material hardening effects. These issues are sometimes difficult to eliminate through simple modifications of the sample or the test design. Finally, it is of interest to make use of the data beyond necking, where the strain state departs significantly from ideal conditions. These metrology issues lend themselves to solution by inverse methods, where full field strain measurements and global load measurement data are available. In this paper we describe typical measurement data and explore methods to identify the constitutive response from dynamic tensile tests.

KEY WORDS: High Strain Rate, Advanced High Strength Steels, Digital Image Correlation, Finite Element Analysis, Virtual Fields Method

INTRODUCTION

High strain rate tensile testing of Advanced High Strength Steels (AHSS) are of interest to automotive manufacturers who need to assess the crashworthiness of these new materials that are being developed to make lighter, more fuel-efficient vehicles. Crashworthiness, in a relative sense, is estimated from dynamic strength and ductility obtained from stress-strain curves measured to failure. Crash performance is analyzed in a more absolute sense by large finite element simulations of full-scale vehicle impact problems, which relies on full stress-strain curves over a range of strain rates, to define the material response up to, but not including, the fracture point. Modeling fracture requires many additional mechanical tests that are not discussed here [1].

In the past several years it has become apparent that traditionally-designed dynamic tensile tests, performed using Kolsky Bar methods can produce flawed results. High speed Digital Image Correlation (DIC) measurements show that the strain experienced in specimens can deviate markedly from what is indicated using traditional Kolsky bar data analysis methods [2]. These deviations can be manifest in a variety of ways. Typically the strain measured by DIC methods in the gage section is substantially less than indicated by Kolsky bar data analysis, owing to some amount of plastic strain occurring outside the nominal gage section. In addition, strain non-uniformities arise when small gage length specimens are used to achieve high overall strain rates, rapid force equilibration and to promote fracture. The ASTM E8 standard for static tensile testing calls for a minimum ratio of gage length to gage width of 8, whereas ISO 26203-1:2010, a standard for dynamic testing, allows for ratios as low as 2. For some materials, particularly high strength ones that have limited strain hardening capacity such as most AHSSs, highly non-uniform strain distributions can result [3]. One way to combat this is to customize the specimen geometry for each individual material. However, this is a daunting prospect and makes it difficult to compare different materials on an equivalent strain rate basis. An alternative is to use so-called inverse methods, which may be employed to provide more accurate estimates of material behavior from non-ideal experiments. In addition, opportunities exist to use inverse methods to extract material information from portions of a Kolsky bar test that traditionally are ignored: the ringup portion of the test, where the forces and strain rates are varying, and the post-necking behavior where tri-axial stresses develop that cannot be analyzed using load-deflection measurements unless approximate Bridgeman-type correction methods are applied [4].
The literature is rich and growing fast on inverse method techniques and applications. Review papers on inverse methods have described different methods considered and benefits and drawbacks of the various approaches [5]. One significant

Abu-Farha, Fadi; Mates, Steven.
"Opportunities for Inverse Analysis in Dynamic Tensile Testing."
Paper presented at Society for Experimental Mechanics 2016 Annual Meeting, Orlando, FL. June 6, 2016 - June 9, 2016.

distinction we note is whether or not finite element analysis (FEA) is performed. The Virtual Fields Method (VFM) is a powerful technique that can identify material parameters directly from DIC strain field data without performing expensive FEA. Instead, parameters can be obtained simply by solving a small linear system of equations for elasticity problems, or by minimizing a cost function for plasticity problems [6]. More recently, the dynamic VFM (dVFM) has been developed that uses acceleration field information derived from DIC displacement field measurements to estimate internal forces [7]. Avoiding expensive FEA computations greatly improves speed and reduces the cost of the analysis, and many practical applications have shown remarkable accuracy when the method is expertly applied. The VFM method is limited, however, to two dimensional problems because the DIC measurements upon which it relies can measure only surface strain fields [6]. FEA-based methods avoid this problem, in that results can be obtained without three dimensional data. Another benefit of using FEA-based inverse methods is that one can use the same numerical approximation method to identify material behavior that is used to solve problems of practical significance. This ability minimizes errors that can arise when numerical approximation methods are mismatched.

In this paper we explore opportunities to apply either VFM or FEA-based inverse methods to better leverage experimental measurements from Kolsky bar strain gage signals and high speed DIC displacement field data to experimental results that may be flawed due to specimen geometry or to portions of the experiments that have not traditionally been used to enhance our understanding of material behavior.

RESULTS

Three aspects of dynamic tensile testing that may lend themselves to inverse methods are considered here. First, an initial "ringup" period exists in the Kolsky bar test where force equilibrium has yet to develop and the data are therefore considered unreliable and ignored. Second, depending on the test geometry and material hardening characteristics, the strain in the gage section can be non-uniform after force equilibrium is established due to end effects. Third, after the onset of localized necking through to failure when uni-axial analysis becomes inappropriate. We note that other localization phenomena exist that may provide opportunities for inverse analysis, such as strain localization due to Portevin-Le Chatelier banding, are not considered here. The three portions of the dynamic tensile test are now considered in order.

Ringup

The ordinary way to treat data during the ringup portion of a Kolsky bar test is to ignore it, on the reasonable grounds that the stress and strain state is highly non-uniform and the strain rate is varying rapidly. These are, however, the exact conditions that are exploited by the dVFM [7], which depends on the presence of significant accelerations within the test piece that can be measured with DIC to provide dynamic force information. Typical high speed digital cameras used for dynamic material measurements, such as the ones used in this study, operate at about 100 000 frames per second (100 kHz) and can measure displacement and strain fields with adequate spatial resolution for a tensile Kolsky Bar test. We also note that force and displacement information provided by Kolsky bar strain gage data must be acquired at 200 kHz or more to be fully resolved [8]. Thus typical high speed cameras may not acquire DIC information fast enough to capture the real acceleration fields to make proper use of the dVFM method. Although there are more advanced cameras that can meet or exceed strain gage measurement rates, they can be quite expensive and may also be limited in terms of the total number of frames that can be captured.

In Figure 1 we compare the acceleration obtained from a three dimensional (3D) surface DIC measurement using typical high speed camera equipment and resolution (90 000 fps, 128x288 pixels, 1 μs exposure, 15 pixel DIC subset with 3 pixel offset) with the acceleration predicted from an analysis of the reflected strain pulse from the bar end, which is recorded at 2 MHz using a high speed oscilloscope. The sample in this test is a QP-980 high strength steel, and the gage section measures 2.9 mm wide by 7 mm long by 1.0 mm thick. The DIC acceleration is averaged over a slice of the sample taken perpendicular to the load axis, located nearest the incident bar where the acceleration is highest during ringup. Further, the accelerations obtained from both DIC field information and strain gage data are smoothed by central differencing. As this figure shows, the DIC acceleration field data differ from the strain gage result nearest the acceleration period during ringup. Some of this difference may be real and caused by slippage between the sample and the grip, which is typical but undesirable in Kolsky bar testing. To compare accelerations between DIC and strain gage data at higher recording rates without grip effects, Figure 2 plots DIC measurements obtained directly on the end of the incident bar, this time at 180 kHz, against the strain gage signal. With grip slippage eliminated there is better agreement between DIC and strain gage results, but we still see larger noise in the DIC data near the peak accelerations compared to the strain gage result. Clearly, obtaining acceleration information from DIC displacement field measurements is quite prone to noise because the data must be differentiated twice, amplifying any displacement noise. Noise levels are quite significant in the 2 MHz strain gage signals that have only been differentiated a single time. Thus even with higher speed cameras for DIC measurements, noise in the derived acceleration data during the ringup portion of a Kolsky bar experiment presents a challenge to obtaining accurate dVFM results. As

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

discussed by its inventors [7], dVFM may be better suited to experiments that are intentionally designed to provide large, smoothly changing accelerations. Turning to FEA-based inverse techniques, the question arises whether data obtained during ringup is useful for identification purposes since in principle the optimized FEA solution would be able to capture this portion of the experiment. Displacement data could be used to guide the identification process so double differentiation is avoided. However, because more constitutive parameters are sensitized during this portion of the test (strain rate sensitivity, for example), the identification process becomes more difficult.

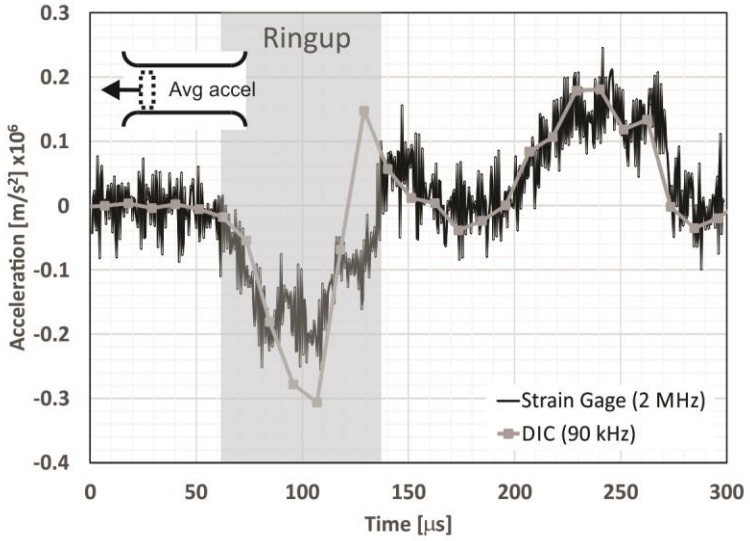


Fig. 1. Comparison of the acceleration of the incident side of a steel tension specimen obtained from DIC measurements at 90,000 frames/s with the acceleration of the incident bar obtained from the reflected strain pulse recorded at 2 MHz. Accelerations are determined by central differencing.

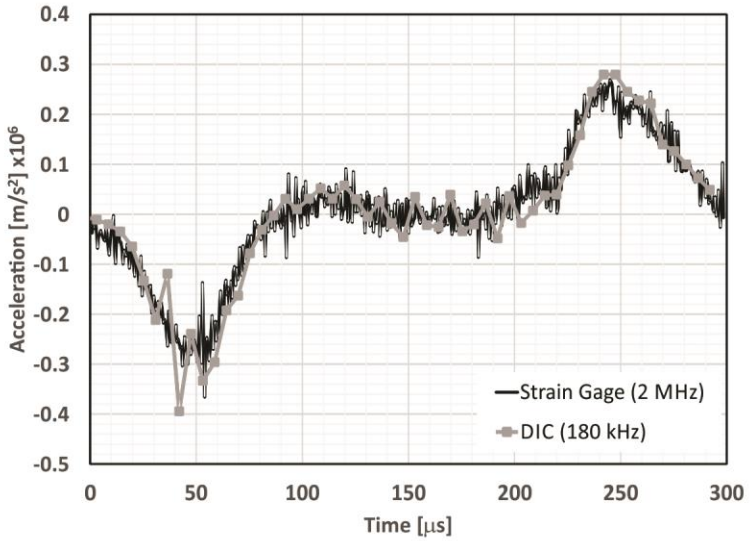


Fig. 2. Comparison of the acceleration of the free end of the incident bar with no sample obtained from DIC measurements at 180,000 frames/s with strain gage results obtained at 2 MHz using central differencing.

Equilibrium Deformation

In Fig. 3 shows dynamic tensile test results for a second-generation AHSS using a gage length of 7 mm and a length-to-width ratio of 1.4. An estimate of the dynamic true-stress, true strain response of this material shows limited strain hardening in the material. High speed 3D DIC measurements using the same settings as the previous data (90 000 fps, 128x288 pixels, 1 μs exposure, 15 pixel DIC subset with 3 pixel offset) show highly peaked true strain profiles across the cross section of the specimen for global average true strains exceeding a few percent. For a material with very little strain hardening such as the QP-980 material studied here, achieving uniform strain along the gage section is very difficult. In the limit of a zero strain

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

hardening material, one can demonstrate with finite element analysis that localized necking begins almost immediately because the material is incapable of diffusing a neck by hardening. In Fig. 4 the same steel is tested with the same gage length (7 mm) but a reduced thickness such that the gage length-to-width ratio is now 2.4. Clearly the strain profile is more uniform, but the strain falls off significantly at the edges even within the gage section. We note that the strain rate is higher in the second test because of reduced cross sectional area of the specimen lowers the transmitted load which, in turn, increases the overall strain and strain rate in the test. Thus even seemingly minor changes to the specimen geometry can impact the test conditions significantly.

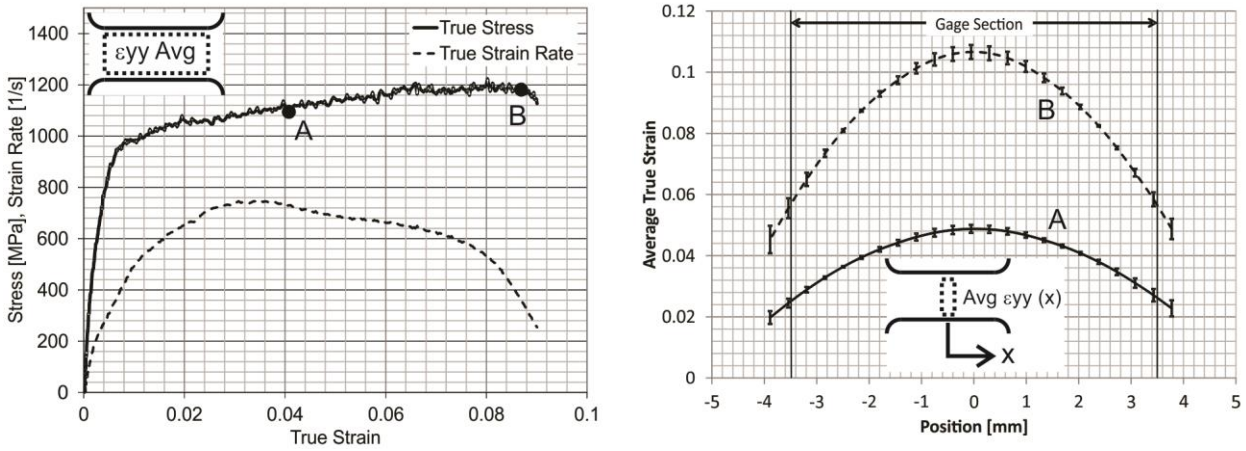


Fig. 3. Left: Dynamic average stress-strain behavior of an AHSS with a gage length-to-width ratio of 1.4. Right: True strain distribution along the load axis at two average true strain values (labeled A and B), from DIC data averaged across planes perpendicular to the load axis.

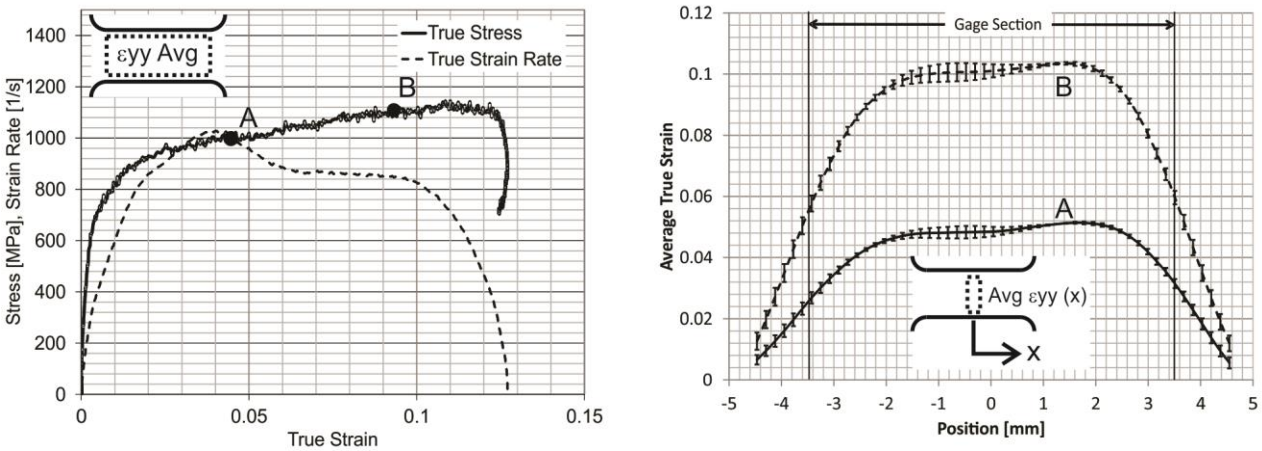


Fig. 4. Left: Dynamic average stress-strain behavior of an AHSS with a gage length-to-width ratio of 2.4. Right: True strain distribution along the load axis at two average true strain values (labeled A and B), from DIC data averaged across planes perpendicular to the load axis.

Equilibrium deformation can be analyzed by the ordinary (non-dynamic) VFM because the strain fields are quite non-uniform, the equilibrated forces acting to cause the strain field are available from strain gage data, and the specimen is assumed to be in a state of plane stress up to the point of localized necking. Again, depending on the assumptions made and the complexity of the plasticity model, plastic parameters may need to be identified by minimizing a cost function rather than solving a set of algebraic equations. Additionally, the dynamic nature of the test introduces an additional complication that is unique to dynamic plasticity experiments: adiabatic heating of the specimen due to rapidly accumulating plastic strain. In this case an additional step is needed to evaluate the temperature field and its effect on the virtual work estimate which affects the resulting identified material parameter values. Accounting for adiabatic heating has been recognized as a significant challenge by the developers of the VFM [6]**.** An iterative updating scheme, combined with an assumption, based on measurement data, regarding the fraction of plastic work converted to heat, might be used to account for adiabatic heating

effects in this case. FEA-based inverse methods can also be applied to this portion of the test, albeit at considerably greater complication and computational expense, and with no obvious advantage over the VFM other than providing consistent numerical approximations between the identification problem and the application problem.

Necking

Once localized necking begins, ordinarily no more information can be obtained from a dynamic tensile test. Acceleration levels are much lower during necking than at the onset of the test except when fracture occurs. In addition, the stress state within the developing neck transitions from plane strain to tri-axial, which by itself precludes the use of the VFM because one can no longer accurately estimate virtual work quantities where stresses vary through the thickness of the sample. Further, because the gradients in strain become large in the neck and, in general, high speed cameras have limited spatial resolution, the strain or displacement field resolution is too limited to accurately capture these gradients. This problem also affects the quality of FEA-based identification techniques but it does not preclude their use to estimate material parameters in the neck region up until the fracture point.

CONCLUSIONS

Measuring the dynamic tensile stress-strain curves of advanced high strength automotive sheet steels presents opportunities for inverse analysis to better leverage experimental data obtained when imperfect test conditions arise due specimen geometry effects that are exacerbated by the relatively low hardening rates in these steels. Three portions of the dynamic test were examined regarding the potential of various inverse techniques to improve the value of the test. The dVFM may be very useful to analyze material behavior during ringup, but DIC data must be obtained at very high sampling rates for accurate results in this kind of experiment. After ringup and before necking, ordinary (non-dynamic) VFM can be applied and should perform well if adiabatic heating effects can be adequately accounted for. Once localized necking begins, finite element based inverse methods must be used because the tri-axial stress state that develops in the neck precludes use the VFM. More costly FEA-based identification methods can be applied to all three portions of the test, although the accuracy of the identified parameters can be limited by the quality of the approximations needed in the absence actual of three dimensional data. FEA-based methods can also be selected to minimize approximation errors between the identification problem and the application problems. It is also noteworthy that the present paper deals only with dynamic tests designed using the traditional approach that seeks a uni-axial strain state and rapid development of force equilibrium. As pointed out in [7], there may be significant benefits to designing material tests very differently in order to maximize the utility of the dVFM method in making use of internal accelerations within the test sample that are intentionally avoided by the traditional materials testing approach.

ACKNOWLEDGMENT / DISCLAIMER

This material is based upon work supported by the Department of Energy under Cooperative Agreement Number DE-EE0005976, with United States Automotive Materials Partnership LLC (USAMP). This support is greatly appreciated. Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] Bao, Y., and Wierzbicki, T. (2004). On Fracture Locus in the Equivalent Strain and Stress Triaxiality Space. International Journal of Mechanical Sciences, 46, 81-98.

[2] Gilat, A., Schmidt, T., and Walker, A. (2009). Full Field Strain Measurement in Compression and Tensile Split Hopkinson Bar Experiments. Experimental Mechanics, 49, 291-302.

[3] Mates, S., and Abu-Farha, F. (2015). Dynamic Tensile Behavior of a Quenched and Partitioned High Strength Steel using a Kolsky Bar. Proceedings of the Society for Experimental Mechanics 2015 Annual Meeting. Costa Mesa, CA: Society for Experimental Mechanics.

[4] Bridgman, P. (1952). Studies in Large Plastic Flow and Fracture. New York: McGraw-Hill.

[5] Avril, S., et al., (2008). Overview of Identification Methods of Mechanical Parameters Based on Full-field Measurements. Experimental Mechanics, 48, 381-402.

[6] Pierron, F., and Grédiac, M. (2012). The Virtual Fields Method. New York: Springer.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

[7] Pierron, F., Zhu, H., and Siviour, C. (2014). Beyond Hopkinson's Bar. Philosophical Transactions of the Royal Society A, 372 (2023), 24.
[8] Chen, W., and Song, B. (2010). Split Hopkinson (Kolsky) Bar. New York: Springer.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

An Investigation of the Temperature and Strain-Rate Effects on UHMWPE Fibers

Donald R. Jenket II, Pathways Fellow, National Institute of Standards and Technology
M/S 8102 100 Bureau Dr. Gaithersburg, MD 20899
Amanda M. Forster, Research Scientist, National Institute of Standards and Technology
M/S 8102 100 Bureau Dr. Gaithersburg, MD 20899
Nick G. Paulter Jr., Group Leader, National Institute of Standards and Technology, 100 Bureau Dr.
M/S 8102, Gaithersburg, MD 20899
Tusit Weerasooriya, Group Leader, Army Research Laboratory
WRMD, Aberdeen Proving Ground, MD 21005
Carey A. Gunnarson, Research Scientist, Army Research Laboratory
WRMD, Aberdeen Proving Ground, MD 21005
Mohamad Al-Sheikhly, Professor, Department of Materials Science and Engineering
University of Maryland 4418 Stadium Dr., College Park, MD 20742-2115

ABSTRACT
During a ballistic impact, materials are subjected to both high strain-rates and high temperatures. Ultra High Molecular Weight Polyethylene (UHMWPE) fibers used in ballistic protection and their strength increases with increasing strain-rate and decreases with increasing temperature. To understand the impact of both factors, a single fiber heater has been fabricated to heat UHMWPE fibers up to their melting temperature (~148 °C) to measure the change in mechanical properties as a function of temperature and strain-rate. Custom grips have been fabricated for use with the single fiber heater and performed well across all strain rates and temperatures in this study. 251 tensile tests have been conducted on 10 mm gage length UHMWPE single fibers at temperature-strain-rate combinations spanning five decades of strain-rates and eleven temperatures. A non-failure boundary is found by fibers that can be strained to 25% without mechanically failing beginning at 75 °C for 10^{-3} s^{-1}, 100 °C for 10^{-2} s^{-1}, 130 °C for 10^{-3} s^{-1}, 148 °C for 10^{-0} $s^{-}1$, and fail regardless of temperature at 550 s^{-1}. It is estimated that an increase in temperature of 25 °C to 30 °C is equivalent to lowering the strain-rate by one decade. At 550 s^{s-1} strain-rate, there was little change in the strain-to-failure from 20°C to 145 °C indicating strain-rate is the dominant factor. Similar constant strains-to-failure are seen in quasi-static tests at 65 °C to 75 °C below the respective non-failure boundary.

Keywords: UHMWPE single fiber, strain-to-failure, Split-Hopkinson Tension Bar (SHTB), Kolsky bar, single fiber heater

BACKGROUND
Ultra-High Molecular Weight Polyethylene yarns exhibit a high strength-to-weight ratio, low density, and are commonly used in rope and body armor applications. The polymeric material is viscoelastic and the mechanical properties have a strong dependence on strain-rate and temperature. During a ballistic impact the material is subjected to high temperatures and high strain-rates. The increase in temperature decreases the strength and increases the strain-to-failure by allowing the chains a higher degree of mobility and increasing the likelihood for chain slippage. The effects from increasing the strain-rate are reversed; as the strain-rate increases, the strength increases and the strain-to-failure decreases since the chains do not have

time to move or slip on such short time scales. Understanding how these competing influences affect the overall mechanical behavior during a ballistic impact is essential for predictive modeling and advances in the design of soft body armor systems.

A single UHMWPE fiber is the smallest discernable component of the soft body armor system and is often tested to extrapolate the mechanical properties to the yarn and other higher-level constituents. Tensile tests are performed by gripping the ends of a single fiber and applying a constant strain-rate until failure. The single fibers have a low surface energy [1] making them inherently slippery and a high tensile strength, causing difficulties in proper gripping of the fibers during tensile testing [2, 3]. Capstan methods have been used to gain data about failure stresses but the failure strains from this method is often questionable due to the unknown amount of gage length within the capstan [2, 4]. Direct gluing to cardboard has shown limited success and is dependent on fiber diameter [5, 6] while a combination of glue and pressing between rubber tabs has reported success but on small numbers of tests performed [3]. Fiber-specific grips have also been developed to directly grip Kevlar fibers between two tabs of poly(methyl methacrylate) [7-9] and the same design using polycarbonate tabs has shown to be successful for UHMWPE singles fibers [10]. The gripping tabs in these designs are recessed from the grip edge making it challenging to access the length of fiber in the grip for heating experiments. To investigate the simultaneous effects of strain rate and temperature on the mechanical properties of UHMWPE single fibers, a new set of grips must be created and compliant with a single fiber heater.

SAMPLE PREPARATION

UHMWPE single fibers were teased from a spool of SK76 yarn. Samples were prepared by using cyanoacrylate to glue ends of a single fiber to a hollow rectangular mounting template cut from an acetate sheet. The hollow rectangle shape allows the 10 mm gage length and the two polycarbonate gripping tabs to fit between the glue points. He mounting template is cut away after the fiber is gripped and before tensile testing. A mounted fiber with one long side of the rectangle removed is shown in figure 1. Fiber diameters were measured using an optical microscope with 60x lens at five points along the 10 mm gage length. The average diameter of fibers used in this study was 18.63 μm with a standard deviation of 1.53 μm. The highest fiber diameter measurement was 23.11 μm and the lowest was 13.86 μm.

EXPERIMENTAL METHOD

A custom single fiber heater and a direct gripping method were used to conduct uniaxial tensile tests on UHMWPE single fibers across five decades of strain-rate and eleven temperatures from room temperature to the melting temperature.

Single Fiber Heater: There were several design requirements for a single fiber heater: the fiber be rapidly heated to minimize artificial annealing of the fibers, the heater maintain a fairly constant and uniform temperature profile along the gage length of the fiber, consistently maintain the same thermal profile between tests, provide a way to accurately measure the real-time temperature of the fiber area, and provide easy access to the fiber channel to allow loading of mounted fiber samples. A design taking into account all of these requirements is shown in Figure 2. This design is for fiber gage lengths of 10mm and the heater thickness is only 9.77 mm to allow the grips clearance from the heater. The top and bottom of the heater are made from Oxygen-free high thermal conductivity (OFHC) copper. The fiber channel is 1.016 mm (40 mil) in diameter. There are guide posts on opposite diagonals to allow the top and bottom to consistently be closed without damaging the fiber. Diagonally symmetric thermocouples are inserted close to the channel center to allow near direct measurement of the channel center. Cartridge heaters are located in the top and bottom halves of the heater. Calibration was achieved by using a butt-welded small diameter wire thermocouple placed in the fiber channel and conducting numerous heating experiments while measuring both the wire thermocouple and thermocouples inside the single fiber heater. The wire thermocouple was placed on an XYZ translation stage to allow precise movement of the thermocouple junction and to determine the thermal profile along the fiber channel. From room temperature, the center of the heater channel reaches 150 °C in 200 seconds. In the two dimensions of the diameter of the channel, the change in temperature from the center to the edge was less than 1 °C for all temperatures. Along the gage length, the change in temperature in the channel from the center to either side of the heater was less than 5 °C at the highest temperature of 150°C. For low strain-rate (10^{-3} s^{-1}) tensile tests, which last several minutes, protocols were developed using the same calibration method to rapidly heat and then hold the channel at temperature ±1 °C by varying the voltage applied to the cartridge heaters. The Heater bottom is mounted on an XYZ translation stage to allow

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

centering of the gripped fiber in the channel, spacing adjustments between grip ends, and raising and lowering for fiber mounting. A stereo optical microscope was used to align and center the fibers in the heating channel.

Custom Grips: Grips were designed using the direct gripping method with polycarbonate tabs as the contacting material [7-10]. The grip design was modified to allow the polycarbonate tabs to extend to the edge of the grip to minimize the gage length outside the heating channel. The polycarbonate was shaped into a "T" to minimize movement of the tabs during tensile testing and the gage length can now be measured from grip-edge to grip-edge.. The grip bottoms are depicted in figure 1. The top and bottom of the heater are made from 303 stainless steel and are pressed together using two 0-80 screws with diagonally placed guide posts to allow for consistent alignment of the polycarbonate tabs and to apply uniform pressure across the gripped portion of the fiber. A precise amount of force is required to properly hold the fiber during tensile tests while being low enough to not induce failure at the fiber-grip interface at the edge of the grip. This force was determined by conducting a series of tensile experiments on single fibers and using a HIOS CL-2000 torque screwdriver to vary the torque placed on the 0-80 screws.

Quasi-Static and Intermediate Strain-rate Experiments: A Bose Electroforce 3100 was used to conduct uniaxial tensile experiments on UHMWPE single fibers at strain-rates of 10^{-3}, 10^{-2}, 10^{-1}, and 10^{0} s^{-1}. A thermal standoff was placed between the grip and the load sensor to prevent thermal effects on the force values. Fibers were heated to the desired temperature and then held within ±1 °C for the duration of the test. After the test, the heater top was removed to allow observation of fiber length protruding from each fiber-grip interface. If both grips presented fiber the event was recorded as a successful failure in the gage length. Slack was determined by comparing the start of the displacement and load curves and added to gage length for calculations.

Dynamic Strain-rate Experiments: A fiber-Split-Hopkinson Tension Bar was used to conduct uniaxial tensile experiments on UHMWPE single fibers at and average strain-rate of 550 s^{-1}. An optical setup was used to measure displacement and a dynamic load sensor was used to measure force [10]. A thermal standoff was also used to prevent thermal effects from affecting the force values. Figure 3 shows the high strain-rate setup. For stress-strain calculations at these high strain-rates, the time delay between the displacement and load curves must take into account the time required for the impact wave to travel through the thermal standoff. An impact hammer with aluminum striker was used to measure the time between impacts at the grip the load sensor. The delay was determined to be 52.3 microseconds which translated the start of the force curve to coincide with the start of displacement curve. Post-test recording was conducted consistent with the lower strain-rate experiments.

RESULTS AND DISCUSSION

Breaks in Gage Length: The custom grips performed successfully across all temperatures and strain-rates. A total of 251 tensile experiments were completed in this study spanning 55 different temperature-strain-rate combinations. Of the 251 tests, only six had failures at the fiber-grip interface. 66 fibers did not mechanically fail (non-failures) and were strained to the machine maximum (25%).The numbers of fibers pulled by strain-rate and temperature are summarized in table 1. The highlighted region indicates the temperature-strain-rate combinations where the fibers begin to have non-failures. The number of fibers pulled by strain-rate and the number of breaks at the grip interface and the comparison to Sanborn et al. are summarized in table 2. Four of the six interface failures occur at the high strain-rate and suggests that there is a strain-rate threshold where the success rate for the direct gripping method begins to decrease. Due to the high success rates seen in this study, it is difficult to draw further conclusions but this does support the findings of Sanborn et al. regarding this phenomenon. The six fibers that failed at the fiber-grip interface occurred between 20 °C and 65 °C. Five of the six interface failures have breaking strengths above the average for the samples in that temperature-strain-rate. The last one of the six was at 10^{-1} and 35 °C and was measured to have an ultimate tensile strength of 3.36 GPa while the average for that group was 3.4 ± .16 GPa. This sample is only 1.5% below the average strength while being well within the standard deviation for the rest of the group. Overall, the grips performed well and have shown a drastic improvement in rate of success over previously reported grips at high strain-rates.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

The non-failures are temperature and strain-rate dependent beginning at 75 °C for 10^{-3} s^{-1}, 100 °C for 10^{-2} s^{-1}, 130 °C for 10^{-3} s^{-1}, 148 °C for 10^{-0} s-1, and are not observed for 550 s^{-1}. These regions indicate areas where the conditions exist for a near steady-state of thermally-activated chain motion (temperature) and time (strain-rate). Additionally, a qualitative estimate can be made from the step-shape of the non-failure regions in table 1that for similar mechanical response an increase in temperature of approximately 25°C to 30 °C is equivalent to raising the strain-rate by one decade.

Strain-to-Failure: Strain-to-failure is another metric for the performance of the custom grips. ASTM C1557-03 (2008) requires strains to be compliance corrected using plots of the displacement to failure, force at failure, gage length and cross sectional area of the fiber to determine the instrument compliance [11]. This method assumes a brittle fiber that has a linear-elastic stress-strain response until failure. The method is not ideal for viscoelastic materials such as UHMWPE fibers which behave non-linearly due to the presence of viscoelastic creep that increases as the fibers are heated, strained at lower strain-rates, or both. Determination of the linear portion of the stress-strain curve for Young's Modulus calculations is also challenging under these conditions [10]. Additionally, necking is observed for fibers starting at 50 °C for 10^{-3} s^{-1}, 50 °C for 10^{-2} s^{-1}, 85 °C for 10^{-3} s^{-1}, 115 °C for 10^{-0} s-1, but does not appear for 550 s^{-1}. For conditions where necking is present, the ultimate tensile strength (UTS) has a higher value than the failure stress and this is not accounted for in the standard. Considering these challenges to calculating compliance for UHMWPE single fibers, the best linear fit for compliance of the Bose Electroforce 3100 came from the 20°C 10^{-3} data where necking was absent. The compliance should not change significantly between single fiber experiments and this value of compliance was used to calculate corrected strain-to-failures for all temperatures and strain-rates using this instrument. Similarly, the best linear fit for the compliance of the fiber Split-Hopkinson Tension Bar was using the data for 50 °C and all strain-to failures were corrected with this value of compliance. Table 3 lists the average uncorrected strain-to-failure of the fibers at the temperatures and strain-rates and table 4 lists the average corrected strain-to-failure. Both the uncorrected and corrected values at 20 °C show lower strains-to-failure averages for 10 mm gage length samples at each strain-rate compared to the values of Sanborn et al. and the comparison is shown in table 5. The corrected values agree with the manufacturer's values for strain-to-failure of 3 to 4% for similar fibers [12]. These comparisons at room temperature indicate the grips are performing well and that the strain-to-failure values at higher temperatures are valid.

The data also quantitatively supports the qualitative estimate for the equivalence of approximately 25°C to 30 °C to one decade of strain-rate. The average strain-to-failure values in the quasi-static strain-rates have similar values when shifted relative to the non-failure boundary. For example, the average corrected strain-to-failures for the lowest three strain-rates are in the approximately 13% at 10 °C to 15°C below the boundary, approximately7% at 25 °C to 30°C below the boundary, and in the high 4% range at three temperature-zones below the boundary (40 °C to 45 °C).

The 550 s^{-1} corrected average strain-to-failure did not change significantly as temperature increase. At this strain-rate, the thermally-induced mobility of the chains is negligible in comparison to the time scale the strain-to failure remained around 3% for all temperatures excluding the 148 °C experiments which are just slightly lower. This behavior is observed in the quasi-static regime at 65 °C to 75 °C below the non-failure boundary. The strain-to –failure is approximately 3.15% in the temperature range between 20 °C to 75 °C at 10^{0}, 20 °C to 50 °C at 10^{-1}, and 20 °C to 30 °C at 10^{-2}. Lower experimental temperatures are needed to confirm this behavior at 10^{-3}. The data provides insight into the molecular dynamics as a function of temperature and strain-rate and supports shows promise to support time-temperature-superposition models when shifted relative to the non-failure boundary.

CONCLUSIONS

Dyneema SK76 UHMWPE single fibers with 10 mm gage lengths were successfully gripped and pulled in uniaxial tension tests over 5 decades of strain-rate and eleven temperatures from room temperature to the melting temperature using a new single fiber heater and custom grips. The new grips demonstrated a high rate of success across all strain-rates and showed lower average strain-to-failure values at room temperature compared to previous reports [3, 10]. Due to the successful performance of the grips, a high level of confidence is given to the observed changes in mechanical properties from the effects of elevated temperatures using the single fiber heater. The stress strain curves indicate a change from viscoelastic

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

"brittle" to ductile response as temperature increases for a given strain-rate. A necking of the fibers begins at 50 °C for 10^{-3} s^{-1}, 50 °C for 10^{-2} s^{-1}, 85 °C for 10^{-3} s^{-1}, 115 °C for 10^{-0} s^{-}1, and is absent for 550 s^{-1}.The strain-to-failure decreases with increasing strain-rate and increases with increasing temperature. Fibers can be strained to 25% without mechanically failing beginning at 75 °C for 10^{-3} s^{-1}, 100 °C for 10^{-2} s^{-1}, 130 °C for 10^{-3} s^{-1}, 148 °C for 10^{-0} s^{-}1, and fail regardless of temperature at 550 s^{-1}. This non-failure boundary provides an estimate that for similar strain-to-failure values, increasing the temperature by 25 °C to 30 °C is equivalent to decreasing the strain-rate by a decade. This estimation remains valid in the quasi-static regime until the non-failure boundary reaches the melting temperature which occurs at the 10^{0} s^{-1} intermediate strain-rate. The 550 s^{-1} maintained a nearly constant strain-to-failure of 3% up until the melting temperature. This behavior was observed in the quasi-static regime at 65 °C to 75 °C below the respective non-failure boundary and suggests time-temperature superposition models can be applied when shifted relative to the non-failure boundary.

FUTURE WORK

Conducting tests at additional temperatures and using a quasi-static UTM that could strain the high temperature fibers to failure might help elucidate the non-failure boundary and allow testing of time-temperature superposition models. Additionally, investigating the strength of UHMWPE fibers as a function of temperature and strain rate will be a topic of research. The difficulty in calculating compliances according to ASTM C155-03 (2008) for instruments when using viscoelastic materials indicates a need for the development of a testing standard for polymeric single fibers. Conducting additional tensile tests would help to better fit the compliance and reduce error bars in the data. Lastly, UHMWPE has a relatively low melting temperature and scaling the single fiber heater and the grips to be able to investigate the temperature and strain-rate effects of other ballistic single fibers materials would provide a comparison between material types.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

REFERENCES

1. Lin, S. P.; Han, J. L.; Yeh, J. T.; Chang, F. C.; Hsieh, K. H. Surface Modification and Physical Properties of Various UHMWPE Fiber Reinforced Modified Epoxy Composites. *Journal of Applied Polymer Science* **2007,** *104,* 655–665.
2. Umberger, P. D. Characterization and Response of Thermoplastic Composites and Constituents. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2010.
3. Russell, B. P; Karthikeyan, K.; Deshpande, V. S.; Fleck, N. A. The High Strain-rate Response of Ultra High Molecular Weight Polyethylene: From Fibre to Laminate. *International Journal of Impact Engineering* **2013,** *60,* 1–9.
4. Schwartz, P.; Netravali, A.; Sembach, S. Effects of Strain-rate and Gauge Length on the Failure of Ultrahigh Strength Polyethylene. *Textile Research Journal* **1986,** *56* (8), 502–508.
5. Hudspeth, M.; Nie, X.; Chen, W. Dynamic Failure of Dyneema SK76 Single Fibers Under Biaxial Shear/Tension. *Polymer* **2012,** *53,* 5568–5574.
6. Rosso, S. Del, Iannucci, L., Curtis, P. T., Impact, T., & Method, F. E. (2012). Investigation of novel hybrid braids for impact, (June), 24–28.
7. Kim, J. H.; Heckert, A. N.; Leigh, S. D.; Rhorer, R. L.; Kobayashi, H; McDonough, W. G.; Rice, K. D.; Holmes, G. A. Statistical Analysis of PPTA Fiber Strengths Measured Under High Strain-rate Condition. *Composites Science and Technology* **2014,** *98,* 93–99.
8. Kim, J. H.; Heckert, N. A.; McDonough, W. G.; Rice, K. D.; Holmes, G. A. Single Fiber Tensile Properties Measured by the Kolsky Bar Using a Direct Fiber Clamping Method. In *Proceedings of Society for Experimental Mechanics Conference*, Lombard, IL, 3–5 June 2013.
9. Kim, J. H.; Heckert, N. A.; Leigh, S. D.; Kobayashi, H.; McDonough, W. G.; Rice, K. D.; Holmes, G. A. Effects of Fiber Gripping Methods on the Single Fiber Tensile Test: I. Non-Parametric Statistical Analysis. *Journal of Materials Science* **2013,** *48,* 3623–3673.
10. Sanborn, B., Dileonardi, A.M., Weerasooriya, T.: Tensile properties of Dyneema SK76 single fibers at multiple loading rates using a direct gripping method; ARL-TR-6974; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, (June 2014).
11. ASTM Standard C1557-03, 2008, "Standard Test Method for Tensile Strength and Young's Modulus of Fiber," ASTM International, West Conshohocken, PA, (2008).
12. Dyneema Comprehensive Fact Sheet. CIS YA100, DSM Dyneema LLC, Stanley. http://issuu.com/eurofibers/docs/name8f0d44 (2008). Accessed 10 Feb 2016.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

TABLES

Table 1. Number of fiber samples tested at different temperature and strain-rate combinations, with highlighted areas indicating regions where the fibers are strained to 25% without failing

	Temperature [°C]											
Strain-rate [s^{-1}]	**20**	**35**	**50**	**65**	**75**	**85**	**100**	**115**	**130**	**145**	**148**	**Total**
10^{-3}	4 (1)	5	5	5	1/4*	5	5	5	5	5	4	**54**
10^{-2}	5	5	5	3	3	4	2/2*	4	5	5	5	**48**
10^{-1}	5	4 (1)	4	4	4	4	4	3	5	4	3	**45**
10^{0}	5	4	4	5	5	4	5	4	2	3	3/2	**46**
550	5 (1)	5	4 (1)	3 (2)	6	6	5	5	6	4	5	**58**
Total	**26**	**24**	**23**	**22**	**23**	**23**	**23**	**21**	**23**	**21**	**22**	**251**

*Number of failures in gage length (number of failures at the fiber grip interface)

**Number of failures/Number of non-failures

Table 2. Success rate of the grips at different strain-rates

Strain-rate [s^{-1}]	Number of Fiber Samples	Breaks at Grip Interface	% Breaks in Gage Length	% Breaks in Gage Length by Sanborn et al.
10^{-3}	54	1	98.15	90
10^{-2}	48	0	100.00	-
10^{-1}	45	1	97.78	-
10^{0}	46	0	100.00	91
550	66	4	93.94	-
775	-	-	-	**42**
Total (Excluding non-failures)	**185**	**6**	**96.76**	-
Total (Including non-failures)	**251**	**6**	**97.61**	-

Table 3. Uncorrected Strain-to-failure Averages

	Temperature [°C]										
Strain-rate [s^{-1}]	**20**	**35**	**50**	**65**	**75**	**85**	**100**	**115**	**130**	**145**	**148**
10^{-3}	4.49±.16	5.01±.56	7.20±.69	13.74±5.75	24.87±-	-	-	-	-	-	-
10^{-2}	4.15±.27	4.15±.32	4.75±.20	5.43±.51	7.84±.54	14.40±3.03	20.75±5.90	-	-	-	-
10^{-1}	4.29±.07	4.30±.15	4.02±.29	4.25±.30	4.71±.07	5.66±.43	7.92±.87	14.09±4.86	-	-	-
10^{0}	4.25±.18	4.05±.14	4.16±.15	4.27±.17	4.21±.18	4.26±.17	4.53±.15	4.77±.29	4.66±0	4.42±.86	3.99±.66
550	3.86±.79	4.30±.54	3.76±.44	3.89±.15	3.66±.46	3.74±.51	3.80±.52	3.87±.80	3.48±.46	3.64±.59	2.79±.26

Table 4. Corrected Strain-to-failure Averages

	Temperature [°C]										
Strain-rate [s^{-1}]	**20**	**35**	**50**	**65**	**75**	**85**	**100**	**115**	**130**	**145**	**148**
10^{-3}	3.52±.17	4.24±.48	6.57±.74	13.24±5.75	24.55±-	-	-	-	-	-	-
10^{-2}	3.23±.29	3.14±.39	4.00±.42	4.92±.47	7.24±.48	13.98±2.94	20.23±5.93	-	-	-	-
10^{-1}	3.20±.25	3.18±.13	3.22±.30	3.36±.23	3.84±.02	4.88±.51	7.34±.87	13.70±4.86	-	-	-
10^{0}	3.14±.25	3.11±.15	3.12±.19	3.29±.19	3.25±.12	3.41±.18	3.70±.18	4.13±.31	4.19±.88	4.13±.87	3.73±.62
550	2.97±.76	3.32±.54	2.87±.43	2.88±.12	2.80±.37	2.91±.50	3.03±.47	3.16±.76	2.85±.43	3.12±.60	2.67±.28

Table 5. Comparison of Uncorrected and Corrected Average Strain-to-failure Values for 10 mm Samples at 20 °C

Strain-rate [s^{-1}]	**Uncorrected Strain-to-failure [%]**	**Uncorrected Strain-to-failure Sanborn et al. [%]**	**Corrected Strain-to-failure [%]**	**Corrected Strain-to-failure Sanborn et al. [%]**
10^{-3}	4.49±.16	5.53±.87	3.52±.17	3.93±.96
10^{-2}	4.15±.27	-	3.23±.29	-
10^{-1}	4.29±.07	-	3.20±.25	-
10^{0}	4.25±.18	4.83±.72	3.14±.25	3.35±.25
550	3.86±.79	-	2.97±.76	-
775	-	3.71±.26	-	3.00±.24

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

FIGURE CAPTIONS

Fig. 1 Single fiber sample placed on the open grips for a high strain-rate tensile experiment before the grip tops are placed and the second half of the mounting template is cut away

Fig. 2 Open single fiber heater showing the location of the fiber channel, thermocouples, cartridge heaters, and guide posts

Fig. 3 Fiber-Split-Hopkinson Tension Bar setup showing a loaded fiber before the fiber mounting template is cut away and before the heater top is applied

Fig. 4 Three representative types of stress-strains curve in this study: (a) Stress vs Corrected Strain at 10^{-3} s^{-1} and 20 °C, showing a viscoelastic "brittle" failure. UTS (red) occur at the same point; (b) Stress vs Corrected Strain at 10^{-3} s^{-1} and 50 °C, showing a UTS (red), necking region, and failure; (c) Stress vs Corrected Strain at 10^{-3} s^{-1} and 65 °C, showing a UTS (red), long necking region, and viscoelastic creep after 25% strain

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

A comparison of strain calculation using digital image correlation and finite element software

D Banerjee[1] and MA Iadicola[1]
[1]Material Measurement Laboratory, NIST, Gaithersburg, Maryland, USA
E-mail: Dilip.Banerjee@nist.gov

Abstract. Digital image correlation (DIC) data are being extensively used for many forming applications and for comparisons with finite element analysis (FEA) simulated results. The most challenging comparisons are often in the area of strain localizations just prior to material failure. While qualitative comparisons can be misleading, quantitative comparisons are difficult because of insufficient information about the type of strain output. In this work, strains computed from DIC displacements from a forming limit test are compared to those from three commercial FEA software. Differences between manually, DIC, and FEA calculated strains are assessed to determine if the scale of variations seen between FEA model and experimentally measured DIC strains constitute behavior differences or just numerical differences in the strain calculation methods used.

1. Introduction

Digital image correlation (DIC) data are being extensively used for many forming applications including constitutive law calibration, benchmark calibration, and for comparisons with finite element analysis (FEA) simulated results. The most challenging comparisons are often in the area of strain localizations just prior to material failure. This is because limit strains produce an inhomogeneous strain field prior to imminent failure. There is not a consensus on what constitutes “raw” data in DIC measurement and how estimates of errors and uncertainties in “raw” measurands (e.g., shape or displacement) affect the comparison of DIC to FEA. Smoothing related to DIC analysis parameters used for matching and strain calculation can affect the limit of spatial resolution and appropriateness of comparison to FEA [1]. For example, strain measurement uncertainty increases when the virtual gauge length (over which strain is ascertained) decreases [1]. While qualitative comparisons can be misleading, quantitative comparisons are difficult because of insufficient information about the type of strain output from FEA software. This is because FEA software often do not clearly explain how exactly strains are computed for a given type of element. Additionally, each software computes strain differently for a given type of element i.e. shell etc. In order to understand this, three benchmark problems are constructed, and the results obtained from three commercial FEA software are compared with manual computations (programed in GNU Octave) using linear shape functions and known displacements at nodes. Finally, strains computed from DIC displacements from a Marciniak [2] forming limit test (just prior to failure) are compared to those from three commercial FEA software. Quantitative differences in strains are assessed to determine if the scale of variations seen between FEA and DIC strains constitute behavior differences or just numerical differences in strain calculation methods used.

2. Overview of DIC system and strain calculation procedure

Although strains are frequently reported, the basic level of output from stereo digital image correlation measurements is an array of 3D initial shape positions (X,Y,Z) and the associated array of displacements (U,V,W) in that same coordinate system. In the measured Marciniak test reported here, a pair of 5MP CCD grey-scale lab grade cameras with 35 mm focal length lenses are used to acquire at 5 frames/s. The average magnification of the system is approximately 22.4 pixels/mm. Correlations were done over squares of 19 pixels (≈0.85 mm) with a raster step of 7 pixels (≈0.31 mm). The DIC manufacturer software [**3**] calculates strains using neighboring data points and applies a weighted smoothing with a diameter of five step points (≈ 1.25 mm here) centered on the point of interest decaying to a 10 % weight at the edges. Details of the software calculation are not known. To compare the DIC software and typical FEA calculations to manually calculated strains, the DIC data were organized into four node elements. For the manual calculations, the elements are defined in a local (ξ, η) coordinate system where linear shape functions are used to determine the displacement gradients in (ξ, η) space that are transformed back to physical space. Using these gradients, the Lagrangian strains are calculated including the second order terms at any points

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

of interest in the element. From these values the principal strains and true Hencky strains (ε_{xx}, ε_{yy}, ε_{xy}) are calculated at each point of interest. This formulation of strain does not include the effects of an element that is not initially flat. Although the specimen is nominally flat to begin the test, the measured values do not exactly match a flat X-Y plane. To correct for this each element was rotated and unfolded to flat as part of the manual strain calculation. This flattening procedure held the length and relative orientation of two edges and one diagonal of the element constant resulting in some induced small strains in the calculation, on the order of 10^{-6} strain. Calculations for strains at the integration points were done on each individual element using the lower left node as the reference point for rotation and unfolding to flat. Strain at each node was calculated separately on each of the four elements that neighbor that node (after the afore mentioned element rotation), and then the strains were averaged at that point weighted by the area of each contributing element. Missing elements due to lack of data at a node point were not included in the average calculation.

3. **FEA analysis overview and software used in this study – description of elements used**

Three different problems were considered: (A) a 2-element uniaxial (strictly planar) deformation problem, (B) a 2-element by 2-element patch from the high strain band (see Fig. 3), and (C) the entire area near the high strain band at about 0.4 s prior to fracture in Marciniak test (see Fig. 3). FEA study was conducted using three commercially available software: ABAQUS, ANSYS, and LS-Dyna[i] [4-6]. Based on our selection of problems two different types of elements were used, either 4-noded plane strain or 4-noded shell/membrane element. For Problem A plane strain elements were used since there was no variation in Z-coordinate among the nodes and the strains were functions of planar coordinates alone and the out-of-plane normal and shear strains are equal to zero. Plane strain elements are defined in the X–Y plane, in which all loading and deformation occur. Shell elements are used to model structures in which dimension in one direction is significantly smaller than the other dimensions. Typically, shell elements use this condition to discretize a domain by defining the geometry at a reference surface, where the thickness is defined through the section property definition. Traditional shell elements have displacement and rotational degrees of freedom. Membrane elements (and shell elements with a membrane option) are used for Problems B & C. These elements are akin to shell elements and are essentially surface elements that transmit in-plane forces only (no moments) with no bending stiffness. Element formulations vary among commercial software. In this work, efforts were made to use comparable elements in FEA software to solve the three problems. The elements used are: ANSYS (PLANE182 and SHELL181 with membrane option), ABAQUS (continuum plane strain CPE4 and membrane M3D4), and LS-Dyna (Shell element with plane strain and fully integrated option). Note the following pertaining to the FEA calculation: (a) although both reduced and fully integrated elements were used, only results from fully integrated elements are reported here, (b) FEA comparisons with DIC were done at both node and integration points, (c) non-linear geometry option was always turned on, (d) only linear shape function elements were used, and (e) both Lagrangian and Hencky (true) strains were computed but only Hencky strains are reported for brevity.

4. **Benchmark problems and discussion of results**

Problem A, the 2-element model (N_1-N_2-N_4-N_3 and N_4-N_6-N_5-N_3), was constructed using quadrilateral elements (Fig. 1). The left 2 nodes (N_1 and N_2) were constrained in X, and Y directions. A 0.1 mm displacement in the X-direction was applied to nodes N_3 and N_4 and a 0.3 mm displacement in the X-direction was applied to nodes N_5 and N_6. ANSYS Plane182, ABAQUS CPE4 and M3D4, and LS-Dyna Shell (with plane strain option) elements were used to compute the true normal and shear strains at integration points. Note that each fully integrated elements [7] has 4 integration points that are numbered starting from lower left corner and are incremented in counterclockwise manner in each element. Table 1 lists ε_{xx} computed using the procedure described in Section 2 above (hereafter called "manual calculation") and those computed using the three commercial FEA software. The differences between strains obtained with manual calculation and from each software are listed under the "Difference column". Similar results are obtained for the ε_{yy}, and ε_{xy} but are not shown for brevity. It is clear from this table that although all FEA software produce reasonable values, ABAQUS membrane element is the most consistent with the manual calculation. A similar exercise was conducted on a 2 element by 2 element patch from the

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Marciniak test just before onset of failure (Fig. 3), Problem B. This patch is shown in Fig. 2. For this exercise, the displacement values at each node were used as boundary conditions in a simple static analysis. Strains computed at each integration point were compared to those computed manually. In addition,

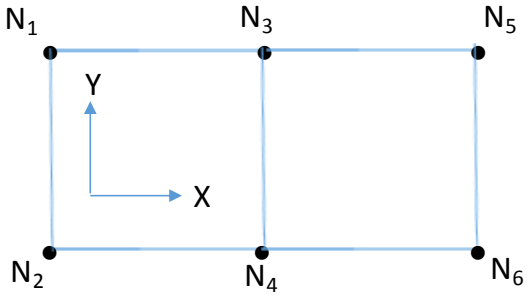


Fig. 1 Problem A.

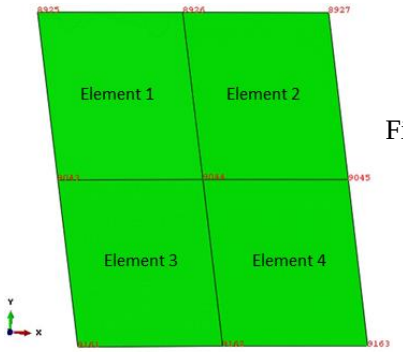


Fig. 2 Problem B.

average nodal strains computed at the common node in the center of the patch are compared. Table 2 shows the true strain values at the integration points. Only results from element 4 are shown here. Again,

Table 1 Problem A ε_{xx} at integration points

	int	Manual	ABAQUS - CPE4		ABAQUS-M3D4		ANSYS-Plane182		LS-Dyna-full integration	
	pnts	calculation	Value	Difference	Value	Difference	Value	Difference	Value	Difference
Element 1	1	0.095348	0.095269	-7.90E-05	0.095349	6.08E-07	0.095142	-2.07E-04	0.095369	1.70E-05
	2	0.095348	0.095269	-7.90E-05	0.095349	8.17E-09	0.095142	-2.07E-04	0.095234	-1.18E-04
	4	0.095349	0.095268	-8.10E-05	0.095349	-3.15E-08	0.095141	-2.08E-04	0.095382	2.90E-05
	3	0.095349	0.095268	-8.10E-05	0.095349	-6.31E-07	0.095141	-2.08E-04	0.095253	-1.00E-04
Element 2	1	0.182323	0.182319	-4.00E-06	0.182323	-4.61E-07	0.181814	-5.09E-04	0.182340	1.70E-05
	2	0.182323	0.182319	-4.00E-06	0.182323	-4.61E-07	0.181814	-5.09E-04	0.182290	-3.30E-05
	4	0.182323	0.182319	-4.00E-06	0.182323	-4.68E-07	0.181814	-5.09E-04	0.182340	1.70E-05
	3	0.182323	0.182319	-4.00E-06	0.182323	-4.68E-07	0.181814	-5.09E-04	0.182310	-1.30E-05

ABAQUS with membrane elements provide the best match to the manual calculation. Both LS-Dyna and ANSYS values are similar and the difference between manual calculation results and values predicted by these software are on the order of 10^{-3}. Note that ε_{xy} are tensor shears in manual calculation which is half the value in vector or Voigt notation in FEA. Table 3 shows the results at the central node, 9044. It is clear that ABAQUS predicted nodal values are closest to manual calculation values. ANSYS predicts slightly larger differences and predicts somewhat better values than LS-Dyna. There are many assumptions in the averaging technique and it is not clear how the nodal strain values are averaged by each commercial FEA software. Finally, the nodal results are compared in Fig. 3 for the actual forming limit test just prior to failure. Here, manual average ε_{xx} are shown in the upper portion of Fig. 3, while the bottom portion shows the difference between this calculation with DIC software (smooth), ANSYS membrane 181, LS-Dyna shell, and ABAQUS membrane elements. In this scale of +/- 0.02 strain difference, the DIC smoothing

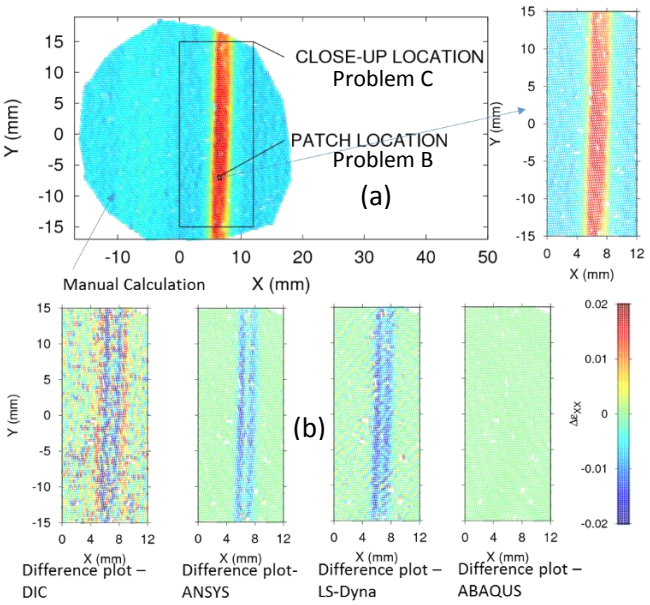


Fig. 3. (a) Manual ε_{xx} and (b) manual and FEA computed ε_{xx} difference plots in a high strain band prior to fracture in a Marciniak test.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

option shows somewhat large strain difference from those obtained with manual calculation (with large differences along the periphery of the band and some random differences over the entire domain). While FEA software used shows better agreement with manual calculation in general (excepting near the high strain band), ABAQUS membrane element provides the best match with the manual calculation. Both ANSYS and LS-Dyna show similar behavior and show strain up to 0.02 strain below the manual calculation at the strain band, which is a somewhat large difference. The present exercise makes the point that reporting of strain values should be made with a better description of how the strains are calculated, what types of elements (along with integration scheme) are used, and what type of averaging scheme is used for obtaining nodal values from element integration point values.

Table 2. Problem B results at integration points.

Element 4		**Manual calculation**		**ANSYS-mem181-full int**			**Difference**		
Int Pt	**ε_{xx}**	**e_{yy}**	**e_{xy}**	**ε_{xx}**	**e_{yy}**	**e_{xy}**	**e_{xx}**	**e_{yy}**	**e_{xy}**
1	0.361250	-0.023102	-0.001579	0.36577	-2.11E-02	2.66E-03	4.52E-03	1.97E-03	5.82E-03
2	0.361250	-0.006998	-0.001845	0.3554747	-5.94E-03	-5.19E-03	-5.78E-03	1.06E-03	-1.51E-03
3	0.361330	-0.004073	0.004426	0.3557066	-6.08E-03	8.29E-03	-5.61E-03	-2.01E-03	-5.60E-04
4	0.361320	-0.020195	0.004531	0.3659433	-2.12E-02	1.64E-02	4.61E-03	-1.03E-03	7.35E-03
				ABAQUS-M3D4-full int			**Difference**		
1	0.361250	-0.023102	-0.001579	0.361250	-0.023102	-0.003144	0.00E+00	2.00E-07	1.37E-05
2	0.361250	-0.006998	-0.001845	0.361253	-0.006998	-0.003672	3.00E-06	-1.00E-08	1.76E-05
3	0.361330	-0.020195	0.004531	0.361335	-0.020195	0.009076	5.00E-06	3.00E-07	1.42E-05
4	0.361320	-0.004073	0.004426	0.361325	-0.004073	0.008869	5.00E-06	2.00E-08	1.71E-05
				LS-Dyna-membrane-full int			**Difference**		
1	0.361250	-0.023102	-0.001579	0.354880	-0.016713	-0.039607	-6.37E-03	6.39E-03	-3.64E-02
2	0.361250	-0.006998	-0.001845	0.350370	-0.001411	-0.041182	-1.09E-02	5.59E-03	-3.75E-02
3	0.361330	-0.020195	0.004531	0.351100	-0.001625	-0.041257	-1.02E-02	1.86E-02	-5.03E-02
4	0.361320	-0.004073	0.004426	0.355600	-0.016928	-0.039682	-5.72E-03	-1.29E-02	-4.85E-02

Table3. Problem B results at central node (9044).

Nodal strains at the central node (9044)							
	Manual	ANSYS-membrane 181		ABAQUS-M3D4		LS-Dyna-membrane	
Strains	Value	Value	Difference	Value	Difference	Value	Difference
ε_{xx}	0.367110	0.367199	-8.948E-05	0.367076	3.35E-05	0.360025	7.08E-03
ε_{yy}	-0.025112	-0.024335	-7.77E-04	-0.025218	1.05E-04	-0.020309	-4.80E-03
ε_{xy}	-0.008190	-0.012267	-4.11E-03	-0.016534	1.54E-04	0.000331	-1.67E-02

5. **References**

[1] Iadicola M A and Creuziger A 2015 *Uncertainties of Digital Image Correlation Near Strain Localizations* Proc. of the 2014 Annual Conf. on Experimental and Applied Mechanics (SEM)

[2] ISO 2008 *Metallic materials-Sheet and strip-Determination of forming-limit curves-Part 2: Determination of forming-limit curves in the laboratory* ISO12004-2:2008(E)

[3] VIC3D (v2010) http://www.correlatedsolutions.com/.

[4] ABAQUS 13.6 software Dassault Systemes http://www.3ds.com/.

[5] ANSYS Mechanical Release12.1 ANSYS Inc. http://www.ansys.com/.

[6] LS-Dyna-971 Livermore Software Technology Corporation http://www.lstc.com/.

[7] Zienkiewicz O C, Taylor R L and Zhu J Z 2013 *The Finite Element Method: Its Basis and Fundamentals* 7th Edition (Oxford: Butterworth-Heinemann)

[i] Certain commercial software or materials are identified to describe a procedure or concept adequately. Such identification is not intended to imply recommendation, endorsement, or implication by NIST that the software or materials are necessarily the best available for the purpose.

Moisture Uptake Characterization in Nanocellulose Using Microwave Cavity

Caglar D. Emiroglu[1,3], Bharath Natarajan[1,3], Jeffrey W. Gilman[1], J. Alexander Liddle[2], Jan Obrzut[1]

[1]Material Measurement Laboratory, [2]Center for Nanoscale Science and Technology,
National Institute of Standards and Technology, Gaithersburg, MD, USA.
[3]Department of Physics, Georgetown University, Washington, DC, USA.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

ABSTRACT

Cellulose nanocrystal films with different helical modulation lengths are characterized. Films are obtained by tuning the rate of evaporation and surface functionalization. Non-contact dielectric measurements are performed using microwave cavity and the water confinement is estimated by employing the classical mixing model with randomly oriented ellipsoidal inclusions. The dielectric constant of absorbed water was found to be significantly smaller than that for free liquid, indicating a limited mobility due to binding with CNC matrix.

Keywords: cellulose nanocrystals, water confinement, dielectric properties.

1 INTRODUCTION

Nanocellulose has been a continuously emerging area of interest over the recent years due to the inexpensive, abundant, and renewable nature of the material source. Cellulose nanocrystals (CNCs) are regarded to be highly useful candidates for a sustainable way of polymer reinforcement. In this regard, investigating the interaction of absorbed water with CNCs is essential to understand its effects on dispersion, wetting, interfacial adhesion, matrix crystallization, and water uptake, all which have critical importance in the utilization and high-scale production of nanocomposites.

Here we consider CNC films cast from water dispersions under controlled drying rate as model structures. For dielectric characterization, a non-contact microwave cavity perturbation method is used [1,2]. The method enables permittivity measurements with adequate accuracy to investigate water confinement in the CNC films. Films are studied via reflectance spectroscopy for measurement of their structural periodicity. In addition, scanning electron microscopy (SEM) is used for imaging, and numerical evaluation of the pitch distributions.

2 EXPERIMENTAL

CNC films were cast from 1.5 % by mass aqueous suspensions inside a sealed Plexiglas chamber. The chamber was supplied with purified air with a relative humidity of 20 % and temperature of 20 °C. Humidity inside the chamber was varied using an evaporation pad with controlled de-ionized water supply. The mass of the CNC suspension, temperature and the relative humidity (RH) inside the chamber were continuously recorded and RH was adjusted according to a computer algorithm that executes the desired drying rate profile. Two drying rates of -7 mg/min and -13 mg/min were employed.

Sulfated CNCs neutralized to a sodium form (Na-CNC) were obtained from the University of Maine Process Development Center. In addition, Na-CNCs functionalized with methyl(triphenyl)phosphonium cations (MePh3P-CNC) were also cast. [3]

The complex relative permittivity, $\varepsilon_r = \varepsilon_r' - j\varepsilon_r''$, of the CNC films was measured at 7.435 GHz utilizing a non-contact cavity perturbation method [1,2], which allows precise measurement of the dielectric permittivity in quantitative correlation with the moisture content. Specimens of CNC films, size of 8 mm × 15 mm and thickness between 30 µm and 60 µm, were cut from the freshly cast films that typically contain 5 % of water by mass. Water concentration was determined gravimetrically by weighing the specimens on a balance with a readability of 100 nanograms. The initial water content was determined by drying films to constant mass at 120 °C for several hours inside of a glove box. By re-exposing CNC to humid air, the water concentration in CNC was varied from about 0 % to about 9 % by mass. At different time intervals during the adsorption, moisture content in the samples was determined gravimetrically, and cavity measurements were performed at room temperature (20 °C).

The total reflectance and transmittance spectra were recorded from 200 nm to 2000 nm with 1 nm step resolution and at slit widths between 4 nm and 10 nm using a Perkin Elmer Lambda 950 Spectrophotometer equipped with an integrating sphere kit. The combined relative uncertainty in determining the helical pitch from the reflection maximum is ± 5 nm.

The optical microscopy images were obtained in a polarized optical microscope (POM) using a 5× objective with numerical aperture NA = 0.13. The images are obtained in cross polarization configuration.

The cross-sections of the dried CNC films were imaged using an SEM at 2 kV accelerating voltage, a current of 50

pA, and pixel sizes varying between 15 nm to 30 nm. A working distance of 4 mm was used.

3 RESULTS AND DISCUSSION

Measured dielectric constant for slow and fast-dried Na-CNC and MePh3P-CNC films are shown in Table 1.

Material	Drying rate	Dielectric constant by mass of water			
		0%	2%	4%	6%
Na-CNC	Slow (-7 mg/min)	3.31	3.69	3.94	4.40
Na-CNC	Fast (-13 mg/min)	3.10	3.30	3.53	3.70
MePh3P-CNC	Slow (-7 mg/min)	2.98	3.05	3.17	3.36
MePh3P-CNC	Fast (-13 mg/min)	2.99	3.05	3.19	3.39

Table 1: Dielectric constant values of CNC films measured using non-contact resonant cavity perturbation method.

The dielectric constant of the slow dried Na-CNC film is considerably larger compared to films dried at faster rates. In comparison, at the corresponding water concentration, the dielectric constant of phosphonium modified MePh3P-CNC films is smaller, and evidently much less dependent on the drying rate. The measured dielectric constant difference between the specimens dried at different rates can be attributed to the degree of chiral ordering in the films, which we discuss in the following sections.

Figure 1 shows the reflectance spectra of Na-CNC and MePh3P-CNC films obtained at different evaporation rates. The reflectance spectrum is found to get consistently narrower and shift to lower wavelengths when the films are cast at slower evaporation rates. It can be observed that continuous monitoring and control of the drying rate, together with the relative humidity and temperature of the ambient, yields films that are self-organized into well-defined and distinguishably different chiral structures with a characteristic pitch (P). Due to the constructive interference between the propagating and back-reflected waves from this periodic structure the CNC films exhibit a selective Bragg reflection [4,5] at $\lambda_{max} = (P/2)\sqrt{\varepsilon_r'}$. Since the dielectric constant of CNC films is about 4 and thus the average refractive index, $n \approx 2$, the helical pitch distribution corresponds directly to the distribution of Bragg reflectance wavelengths.

The films when imaged using polarized optical microscopy displayed multicolored structures in reflectance mode with dominant colors varying based on evaporation conditions and functionalization as shown in Figure 2 together with the corresponding reflectance peak values. The POM images are in agreement with the spectral distributions of the helical pitch P. We also note that the fast dried Na-CNC film was observed to be colorless as it reflects in the near-infrared wavelengths, beyond the visible range.

In order to further investigate the variations in the optical properties arising from differences in evaporation rates as well as from surface functionalization, we studied the structure of the CNC films using SEM.

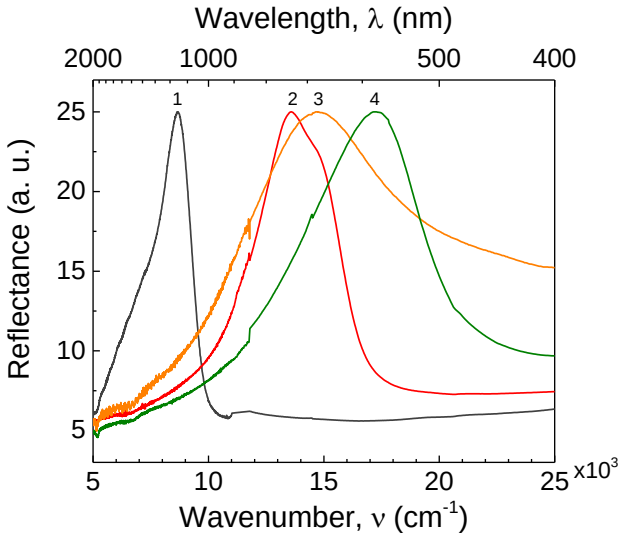


Figure 1: The reflectance spectra of the CNC films by drying rates and functionalization. Na-CNC films: 1 – fast, 2 – slow. MePh3P-CNC films: 3 – fast, 4 – slow. The peak corresponds to the helical pitch, $P \approx \lambda_{max} = 1/\nu_{max}$.

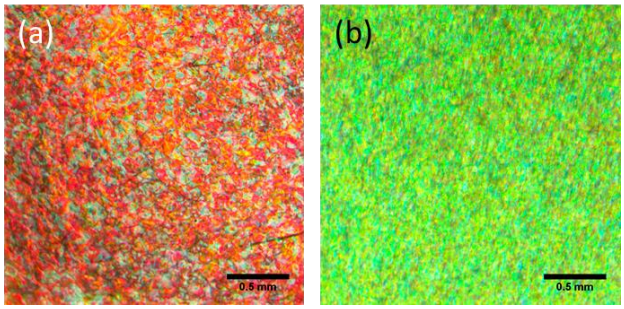


Figure 2: Polarized optical microscope images of (a) Slow dried Na-CNC (P = 736 nm). (b) Slow dried MePh3P-CNC (P = 590 nm). Scale bars represent 0.5 mm.

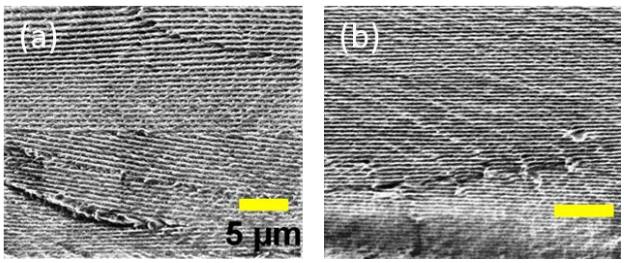


Figure 3: SEM images for (a) Fast dried Na-CNC (P_1 = 1000 nm). (b) Slow dried Na-CNC (P_1 = 606 nm).

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Figure 3 shows the sample SEM images of the cross-section of films. In order to quantitatively evaluate the SEM imaging results, pitch distributions were calculated via image processing. Subsequently, the pitch distributions were fit with Gaussians to obtain peak positions and peak widths (defined at one standard deviation) as shown in Table 2.

Material	Drying rate	P_1 (nm)	ΔP_1 (nm)	P_2 (nm)	ΔP_2 (nm)
Na-CNC	Slow (-7 mg/min)	606	151	-	-
Na-CNC	Fast (-13 mg/min)	1000	350	1500	250
MePh3P-CNC	Slow (-7 mg/min)	504	186	-	-
MePh3P-CNC	Fast (-13 mg/min)	619	190	1045	149

Table 2: The nominal pitch distributions measured from the SEM images of films.

Comparing the reflectance spectra and the SEM results in terms of the peak wavelengths and peak widths, we find that the calculated pitch distributions are in good agreement with the reflectance data. We note that the SEM helical pitch values are somewhat lower than the pitch from the peak reflectance wavelengths. This is attributed to the fact that the reflectance was measured in air as opposed to SEM images, which were obtained in vacuum. In air, we expect swelling from a small amount of absorbed moisture.

Within the assumption that the water uptake into CNC films occurs in between the crystals rather than the swelling individual nanocrystals, the films can be modeled as a two-phase CNC-water mixture. Here we follow the Maxwell Garnett classical mixing approach with randomly oriented ellipsoidal inclusions of water with semi-axial dimensions a_x, $a_y = a_x$, a_z, along x, y, z directions, respectively. [6,7]

The axial parameters of the water inclusions are obtained by fitting the measured dielectric constant of the films against the mixing model with the axial ratio a_x/a_z and the dielectric constant of water being the fitting parameters. The susceptibility ratio is defined by [7]

$$SR = \left(\varepsilon_{eff} - \varepsilon_{CNC}\right)/\left(\varepsilon_{water} - \varepsilon_{CNC}\right) \qquad (1)$$

where ε_{water}, ε_{CNC}, and ε_{eff} are the dielectric constants of the water, the CNC film at 0 % water by mass, and the mixture, respectively.

At our measurement frequency of 7.4 GHz, assuming that the ε_{water} = 65 for free water in the mix [8], the calculated ε_{eff} results in significantly higher values than the experimental data. Consequently, assuming the water inclusions remain spherical for MePh3P-CNC films, where the drying rate has a negligible effect on the dielectric constant (see Table 1), ε_{water} is calculated to be 33. This indicates a strong interaction between water and CNCs. Setting ε_{water} = 33, the axial ratios obtained by fitting the data points to Equation 1 are equal to 30 and 6, for slow-dried and fast-dried Na-CNC films, respectively.

Given that our CNCs differ not only in their mesostructures but also in their affinity to water, we can attribute the water confinement shape in Na-CNCs primarily to intermolecular interaction modulated packing. Na-CNC films from the slow dry process have smaller pitch and therefore tighter packing, which we believe leads to a greater deviation from a spherical shape compared to the fast dried films of lower density and larger pitch. The compression in shape is magnified by the attractive interactions between the hydrophilic Na-CNCs and water. In comparison, in the case of less hydrophilic MePh3P-CNCs, interactions with water are apparently weaker allowing water inclusions to assume spherical shape even in tighter pitch.

4 CONCLUSION

Both the functionalization and the drying rate of aqueous suspensions of CNCs are of critical importance in determining the final film structure, with fast dried films displaying looser packing and larger disorder in long range periodicity as evidenced by SEM images and wider, higher wavelength reflectance peaks.

The microwave cavity pertubation method allowed non-contact dielectric constant measurement of the CNC films. Through an approach using classical mixing models with randomly oriented ellipsoidal water inclusions, it is found that in the case of hydrophilic Na-CNCs, a decreasing pitch led to greater anisotropy in the shape of moisture inclusions (ellipsoidal or platelet-like). In contrast, the structure of hydrophobic phosphonium-cation modified CNC films are found to have little influence on absorbed water inclusions, which remain free and predominantly spherical. These results provide a useful perspective on the current state of understanding of CNCs materials and are beneficial for the realization of CNC functional materials and composites.

REFERENCES

[1] Obrzut, J.; Emiroglu, C.; Kirillov, O.; Yang, Y.; Elmquist, R. E. Surface Conductance of Graphene from Non-Contact Resonant Cavity. Measurement, 87, 146–151, 2016.

[2] Orloff, N. D.; Obrzut, J.; Long, C. J.; Lam, T.; Kabos, P.; Novotny, D. R.; Booth, J. C.; Liddle, J. A. Dielectric Characterization by Microwave Cavity Perturbation Corrected for Nonuniform Fields. IEEE Trans. Microw. Theory Tech., 62, 2149–2159, 2014.

[3] Fox, D. M.; Rodriguez, R. S.; Devilbiss, M. N.; Woodcock, J.; Davis, C. S.; Sinko, R.; Keten, S.; Gilman, J. W. Simultaneously Tailoring Surface Energies and Thermal Stabilities of Cellulose Nanocrystals Using Ion Exchange:

Effects on Polymer Composite Properties for Transportation, Infrastructure, and Renewable Energy Applications. ACS Appl. Mater. Interfaces, 8, 27270–27281, 2016.

[4] Lagerwall, J. P. F.; Schütz, C.; Salajkova, M.; Noh, J.; Hyun Park, J.; Scalia, G.; Bergström, L. Cellulose Nanocrystal-Based Materials: From Liquid Crystal Self-Assembly and Glass Formation to Multifunctional Thin Films. NPG Asia Mater., 6, e80, 2014.

[5] de Vries, H. Rotatory Power and Other Optical Properties of Certain Liquid Crystals. Acta Crystallogr., 4, 219–226, 1951.

[6] Landau, L. D.; Lifšic, E. M.; Pitaevskij, L. P.; Landau, L. D. Electrodynamics of Continuous Media; Course of theoretical physics; 2. ed., and enl.; Elsevier, Butterworth-Heinemann: Amsterdam [u.a], 2008.

[7] Sihvola, A. H. Electromagnetic Mixing Formulas and Applications; IEE electromagnetic waves series; 1st ed. /Reprint with new cover.; Institution of Engineering and Technology: London, 1999.

[8] Hasted, J. B. Aqueous Dielectrics; Studies in chemical physics; Chapman and Hall, New York: London, 1973.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Advances in Next-Generation Polyolefin Standard Reference Materials

Sara V. Orski,[†] Wesley S. Farrell,[†] André M. Striegel,[‡] and Kathryn L. Beers[†]

National Institute of Standards and Technology (NIST)
[†]Materials Science and Engineering Division and
[‡]Chemical Sciences Division
Gaithersburg, MD, 20899; sara.orski@nist.gov

ABSTRACT

NIST's approach to modernizing its synthetic polymer reference materials is to target precise and robust synthetic methods, making homogeneous standards than can be easily tailored to include desired chemistry, molar mass, and topography. This strategy aims to meet the increasing demand for novel standards that are relevant to advanced materials and with minimial development time. Here, an alternative approach to synthetic linear and branched polyethylenes has been developed using a commercial metathesis catalyst to generate polyolefins with low dispersity and sequence control, where the degree of alkyl branching and alkyl branching distribution are known. This process is adaptable to many branch lengths and varied chemistries as it only requires replacement of the monomer feedstock with the desired chemical functionality. The resulting materials are characterized using size-exclusion chromatography with tetra detection to measure the molar mass, molar mass distribution, and degree of short chain branching. In addition to their role as potential standard reference materials (SRMs), these polymers are also being applied to study the effect of branching on polymer chain thermodynamics at surfaces, and comparing experimental results to established theoretical models.

Official contribution of the National Institute of Standards Technology - Not subject to copyright in the United States

Keywords: polyolefins, standards, short-chain branching, molar mass, surfaces

1 INTRODUCTION

The ability to accurately measure polymer molar mass, molar mass distribution, and chemical and topological heterogeneities is paramount to making appropriate structure-property relationships between a material and its desired end-use. All polymers contain some degree of dispersity regarding molar mass. Depending on the chemical and architectural complexity of the polymer, other types of distributions can exist as well, such as chemical composition and short- and long- chain branching. Identifying and quantifying all species of chains present in the material and the resulting impact on material properties is one of the lasting challenges of materials science, despite the rapid progress in polymer science that has occurred over the last 70 years. Several examples of these structure-property relationships are listed in Table 1 [1].

Macromolecular Property	Representative End-use Properties Affected
Molar mass	Elongation, tensile strength
Long chain branching	Shear strength, tack, peel, crystallinity
Short chain branching	Haze, stress-crack resistance, crystallinity
Architecture/Topology	Flow modification, diffusion, encapsulation
Tacticity	Crystallinity, anisotropy, solubility
Chemical Composition	Morphology, miscibility, solubility
Chemical Heterogeneity	Toughness, brittleness, biodegradability
Chemical composition vs. molar mass	elongation, tensile strength, brittleness, toughness, blending, plasticization
Block Sequence	Dielectric properties, reactivity, miscibility

Table 1: Types of macromolecular distributions and their effect on end-use properties of polymers. Reproduced in part from reference [1]. Please see the reference for full list of macromolecular properties and measurement methods.

Measurement of polymers often involves separating samples by fractionation through a porous, packed column, using either enthalpic interaction with a stationary phase (liquid chromatography) or entropic diffusion through a column (size exclusion chromatography (SEC)). SEC fractionates a sample by hydrodynamic size, where the eluting fractions are fully characterized by sensitive physical and chemical in-line detectors to measure the molar mass and the molar mass distribution (MMD). SEC is considered a relative measurement method, since calibration standards of known molar mass are required to calculate number average

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

(M_n) and mass average (M_w) molar mass of the sample. Absolute measurements to determine molar mass exist, such as membrane osmometry to measure M_n and multi-angle static light scattering to measure M_w. These measurements, however, have the disadvantage of long measurement times and larger uncertainties due to manual sample preparation. SEC offers rapid analysis of polymer molar mass with enhanced measurement precision over absolute methods and can even provide more information than absolute methods about the full MMD. Coupling of multi-angle light scattering detection to SEC has also provided the ability to determine "absolute" M_w from a separation, with the assumption that all injected analyte elutes. SEC has therfore become one of most heavily utilized and revolutionary measurements for industrial polymer research and development in the last half-century. Standards development organizations have recognized the need for relative measurements of molar mass and have developed documentary standards for multi-detector SEC to describe best practices for separation and analysis of polymers [2].

Polyolefins, or hydrocarbon polymers generated from petrochemical feedstocks, are the largest volume of plastics produced yearly, at over 130 million tons. The revolution of synthetic processes and novel metal-based catalysts over the last few decades have drastically improved the ability to tailor molar mass, dispersity, and degree of long- and short-chain branching into polyolefins. These capabilities have resulted in polymers with more finely-tunable material properties. To characterize these materials, in-line infrared detectors for SEC have been developed in order to quantify average chemical and branching composition of a polymer across its molar mass distribution curve by measuring methyl and methylene content of the eluting polymer. Standards representative of these complex chemistries and architectures are ultimately required to properly quantify the relationships between molecular structure, processing, and performance of these materials.

NIST's current inventory of polyolefin SRMs were developed beginning in the 1970s to quantify molar mass and melt properties of linear polyethylene and are founded from a common parent material, SRM 1475A, a broad distribution linear polyethylene (M_w = 52,000 g/mol). By fractionating and recrystallizing batches of 1475A, NIST was able to produce six additional narrow MMD standards ranging from 6.28 kg/mol to 196 kg/mol for standards certified to measure intrinsic viscosity, M_n and M_w. These linear standards continue to provide calibration and measurement traceability to stakeholders in academia and industry today, but in time have become less useful to calibrate advanced commercial materials. Specifically, linear polyethylene is not representative of the solution behavior of branched polyolefins and cannot alone be an adequate calibration standard for SEC.

Short chain branching reference materials are available commercially, though in limited supply, and are generally copolymers of ethylene and an α-olefin. These materials, similar to NIST polyethylene SRMs, were isolated by large-scale fractionation to isolate polymers by the degree of short-chain branching. However, fractionating does not address localized dispersity, where different species of the same solubility and hydrodynamic size can co-elute from a column. This phenomenon can potentially lead to masking of small species of non-homogeneous material in the calibration curve and alter the molar mass and solution property measurements.

2 NEXT-GENERATION POLYMER MASS STANDARDS

2.1 Redesigning the Synthetic Approach to Polyethylene

NIST is developing a new approach to modernizing its polyolefin reference materials, which entails the development of new polymer standards essentially correlated to a parent polyethylene species using the same or comparable synthetic processes. These precision polymers, made by controlled polymerization, would minimize uncertainty due to material heterogeneity that is a concern for standards that are fractionated from a heterogeneous mixture (Figure 1).

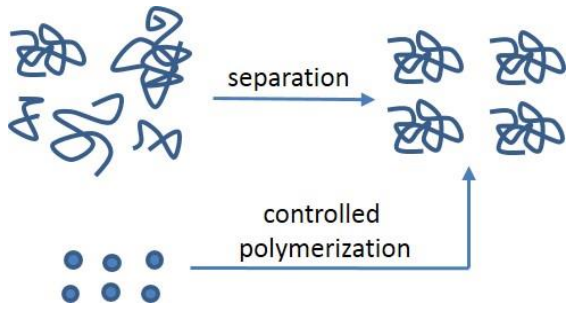


Figure 1: Depiction of homogeneous polymer standards achieved through fractionation (current methods) versus controlled polymerization (our studies).

NIST's approach, shown in Figure 2A, is to use a highly strained cyclic alkene with a commercially available metathesis catalyst to prepare well-controlled polyalkenamers (I.) that can be hydrogenated post-polymerization to polyolefins (II.), a strategy first studied by the Hillmyer group [3]. This has the advantages of making the polymerization accessible to lower temperatures and pressures than commercialized polymerization of ethylene gas and making an initial polymer that is more easily processed and soluble in safer organic solvents. Control of the reaction is achieved by choosing catalyst and polymerization conditions amenable to a fast initiation rate so chain ends initiate simultaneously.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

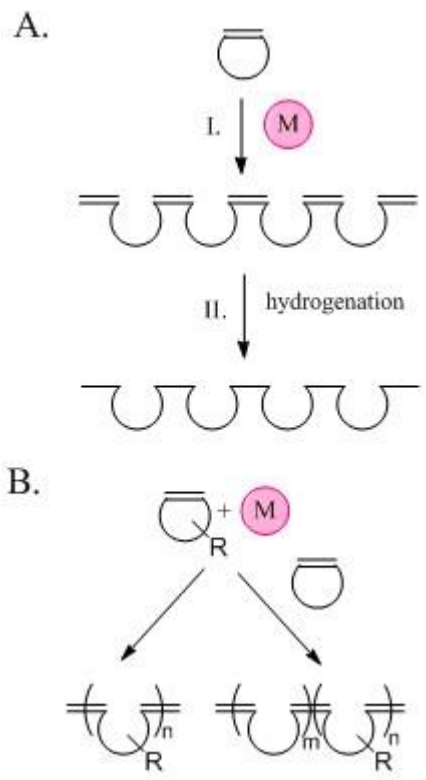


Figure 2: General reaction schemes of (A.) olefin metathesis to synthesize polyalkenamers (I.) with subsequent hydrogenation to polyethylene (II.) and (B.) polymerization of substituted cyclic olefins to generate idealized branched homo- and co-polymers.

2.2 Incorporating Branching and Functionality

The benefit of using a highly active alkylidene catalyst to make polyethylene is that similar polyolefins that incorporate branches and additional functional groups can be easily synthesized by changing the monomer feedstock without altering reaction conditions overall (Figure 2B). Current work at NIST has focused on the incorporation of alkyl branches by careful placement of the alkyl substitution on the ring to take advantage of steric interaction between monomer and the catalyst, and perfect regioregular control of the polyalkenamers was achieved [4]. This ideal head-to-tail addition of monomer places the branch point at a fixed interval along the backbone, yielding an absolute branch frequency and branch length. This material can then be tested as a prototype for a new standard reference material.

2.3 Testing of Linear and Branched Polyethylene Standards

This homogeneous material will undergo extensive testing to quantify molar mass and MMD using both absolute (mulit-angle static light scattering) and relative methods (SEC). These results will be directly compared to SRMs 1484A and 1475A, linear polyethylenes in the same molar mass range, but with varying degrees of dispersity. This comparison ensures that the candidate polyethylene provides equivalent molar mass calibration to these standards, so they may be discontinued once the new SRM is released.

Furthermore, the majority of NIST standards certify either M_n or M_w, which represent only the first and second moments of the molar mass distribution and are not fully descriptive of all molar masses and associated error throughout the entire distribution curve. Analysis of these new standards must address uncertainty across the entire MMD curve.

SEC testing of the linear and branched standards will utilize tetra-detection, including differential refractive index, infrared spectroscopy, viscometry, and multi-angle light scattering, to quantify molar mass distribution and degree of alkyl branching and compare to existing NIST polyethylene standards. The synergistic combination of physical detectors used will also permit measurement of polymer conformational constants, such as the contraction ratio (mean squared radius of a branched versus linear polymer) or the fractal dimension to provide valuble information to NIST customers in addition to certified molar mass values.

3 APPLICATIONS OF ADVANCED POLYOLEFIN MATERIALS

3.1 End-functionalized Polyolefins for Functional Surfaces

The controlled synthesis of polyalkeneamers also is conducive to end-functionalization by reaction quenching with a modified alkene. This allows for reactions of the chain end with a reactive moiety immobilized on a surface. NIST is currently studying grafting of poyalkenamers to a surface before subsequent hydrogenation as a way to stealthily make polyethylene “brush” surfaces under mild solvent conditions, which has not been previously demonstrated. Recent theoretical studies [5] have used Scheutjens-Fleer self-consistent theory models to account for the impact of branching on the conformation entropy loss in polymer brushes in solvents of varying quality. Our studies on linear and branched polyethylenes grafted to the surface will permit experimental testing of that theory as well as permit study of solvent and surface interactions of branched polyolefins near an interface, similar to other work done previously on linear, glassy polymers [6,7].

3.2 Analysis of Asphaltenes

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Asphaltenes are crude oil byproducts that can drastically modify crude viscosity and are largely responsible for the blockage of refinery pipes. The molar mass, MMD, and structure of asphaltenes has not been well-established due to the low critical nano-aggregation concentration (CNAC) observed in organic solvents at low temperatures [8]. Preliminary studies of an asphaltene sample collected indicates a high degree of branching at approximately 150 methyl groups per 1000 carbons, consistent across the molar mass distribution. Concentration studies are underway to determine the CNAC concentration of the asphaltene at 160 °C in 1,2,4-trichlorobenzene to determine accurate solubility conditions for quantifying the molar mass, MMD, and degree of long and short chain branching of the non-aggregated species.

4 CONCLUSIONS

NIST's development of next-generation polyolefin standards is aimed at bridging the gap between existing needs for polyethylene molar mass standards and the immediate industrial need for representative branched polyolefin standards. This approach establishes a systematic process to develop new polyolefin standards with control of monomer molar mass and structure. This infrastructure will ensure that NIST products continue to enable innovation at the forefront of materials science and technology, and provide a manufacturing route to quickly adapt new standards and provide traceable measurements to our stakeholders. This will ensure measurement accuracy remains as high for emerging polymers as the commodity materials that have been characterized for decades.

REFERENCES

[1] Striegel, A. M.; Kirkland, J. J.; Yau, W. W.; Bly, D. D. Modern Size-Exclusion Liquid Chromatography; (2nd ed.), Wiley, 2009.
[2] ISO 16014-(1 to 5) ; ASTM D5296-11
[3] Kobayashi, S., Pitet, L. M. and Hillmyer, M. A. *J. Am. Chem. Soc.*, 2011, 133(15), pp. 5794–5797.
[4] Farrell, W. S.; Beers, K. L. *"Ring-Opening Metathesis Polymerization of Alkyl Substituted Cycloalkenes."* In preparation.
[5] Lebedeva, I. O.; Zhulina, E. B.; Leermakers, F. A. M.; Borisov, O. V. *Langmuir* 2017, 33 (5), 1315–1325.
[6] Sheridan, R.J.; Orski, S.V.; Jones, R. L. Satija, S., Beers, K. L. *"Surface interaction parameter measurement of solvated polymers via model end-tethered chains*." Submitted to *Macromolecules*
[7] Orski, S. V.; Sheridan, R. J.; Chan, E. P.; Beers, K. L. *Polymer* 2015, 72, 471–478.
[8] Dong, S.; Striegel, A. M. *Chromatographia* 2013, 76, 725–733

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Quantitative Analysis of Oxidation State in Cerium Oxide Nanomaterials

Christopher M. Sims, Russell A. Maier, Aaron C. Johnston-Peck, Justin M. Gorham, Vincent A. Hackley, and Bryant C. Nelson

Material Measurement Laboratory, National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA

ABSTRACT

Cerium oxide nanoparticles (nanoceria) are receiving increased attention from the research community due to the large number and wide range of their current and potential applications. The attractiveness of nanoceria for these various applications is rooted in their unique chemical properties, most prominent of which is their ability to alternate between the Ce^{3+} and Ce^{4+} oxidation states. While many analytical techniques and methods have been used to characterize the amounts of Ce^{3+} and Ce^{4+} present (Ce^{3+}/Ce^{4+} ratio) within nanoceria materials, very few studies have utilized multiple complementary analytical tools (orthogonal analysis) with technique-independent oxidation state controls for quantitative determinations of the Ce^{3+}/Ce^{4+} ratio. Use of technique-independent control samples should improve the comparison of oxidation state measurements across a wide range of analytical techniques. Here, we utilize electron energy loss spectroscopy (EELS) and X-ray photoelectron spectroscopy (XPS) to orthogonally characterize the oxidation states of a suite of commercially available nanoceria materials using technique-independent Ce^{3+} and Ce^{4+} controls. Similarities and differences between analytical results are discussed in the context of the products analyzed.

Keywords: ceria nanoparticles, characterization, oxidation state, orthogonal analysis

1. INTRODUCTION

Of the many engineered nanomaterials being incorporated into consumer products, cerium oxide nanoparticles (nanoceria) are receiving increased attention due to their current and potential use in a wide variety of applications.[1] While the performance of nanoceria in these applications depends on many factors, the ability of nanoceria to cycle between the Ce^{3+} and Ce^{4+} oxidation states has been proposed as the primary feature behind their unique abilities.[2] As such, accurate determination of the Ce^{3+}/Ce^{4+} ratio can significantly improve our understanding of nanoceria properties and interactions across a breadth of fields.

In the literature, several analytical techniques have been used to gain insight into the Ce^{3+}/Ce^{4+} ratios of nanoceria (e.g. electron energy loss spectroscopy (EELS), X-ray photoelectron spectroscopy (XPS), Raman spectroscopy), yet each of these techniques operates under different fundamental principles, and hence, are subject to producing different results depending on the technique utilized.[3] However, few studies have used multiple complementary analytical techniques, herein described as orthogonal analysis, for determining the Ce^{3+}/Ce^{4+} ratios in nanoceria analytes. Even in the few cases where orthogonal analysis was performed, technique-independent control samples of known oxidation state were not utilized,[4-5] which further complicates comparison of experimental results between individual analytical techniques.

Here, we describe the development of an analytical procedure designed to measure the cerium oxidation state in nanoceria using orthogonal approaches. Preparation of materials for control measurements and methods for optimizing data acquisition and processing were developed to efficiently analyze and objectively interpret the distribution of Ce^{3+} vs. Ce^{4+} oxides using EELS and XPS. The methodology is applied to quantify the Ce^{3+}/Ce^{4+} ratios in commercially available nanoceria materials.

2. MATERIALS AND METHODS[1]

2.1 Materials

Complete details of materials used and synthesis methods are described elsewhere.[6] Bulk cerium (IV) oxide (CeO_2, 99.995 % Ce) was purchased from Strem Chemicals, Inc. (Newburyport, MA). Cerium carbonate hydrate ($Ce_2(CO_3)_3 \cdot xH_2O$, 99.999 %) and germanium oxide (GeO_2, 99.999 %) were purchased from Alfa-Aesar (Haverhill, MA). Aluminum oxide (Al_2O_3, 99.99 %) was purchased from Johnson and Matthey (Royston, UK). 2 % Ge-doped $CeAlO_3$ (Ge-$CeAlO_3$) was prepared from Al_2O_3, $Ce_2(CO_3)_3 \cdot xH_2O$ and GeO_2 using traditional mixed oxide synthesis techniques modified from previously used methods.[7] Two commercially available nanoceria materials were obtained: a nanopowder comprised of vendor specified 25 nm primary particles (NPCO) and a fuel borne catalyst for

NIST Disclaimer: [1]Certain trade names and company products are mentioned in the text or identified in illustrations in order to adequately specify the experimental procedure and equipment used. In no case does such identification imply recommendation or endorsement by National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

Gorham, Justin; Hackley, Vincent; Johnston-Peck, Aaron; Maier, Russell; Nelson, Bryant; Sims, Christopher.
"Quantitative Analysis of Oxidation State in Cerium Oxide Nanomaterials."
Paper presented at Technical Proceedings of the 2017 TechConnect World, including the Nanotech 2017 Conference, Washington, DC. May 14, 2017 - May 17, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

diesel (FBC). All materials were used as received without further purification.

2.2 Scanning Transmission Electron Microscopy (STEM) Imaging and Electron Energy Loss Spectroscopy (EELS)

High-angle annular dark field images and EELS data were collected using a probe-corrected FEI (Hillsboro, OR) Titan transmission electron microscope. The instrument was operated at 300 kV with a 25 pA beam current. For EELS acquisition the convergence and collection semi-angles were approximately 3.6 and 13.7 mrad, respectively. The dispersion was 0.05 eV/ch, and the FWHM of the zero-loss peak was 0.70 eV ± 0.05 eV. During the spectral acquisition the beam was rastered over an area of 0.026 μm^2, this was repeated in 6 or 7 different locations on the sample grid. To ensure a precise energy-loss scale, a custom Digital Micrograph script switched the drift tube excitation between the zero-loss region and the core-loss region as a series of spectra were recorded. In post-processing spectra were then aligned relative to the zero-loss peak and summed together.

Ge-$CeAlO_3$ and bulk CeO_2 were used as controls to provide the characteristic Ce^{3+} and Ce^{4+} spectra, respectively. A GSL least squares fit, as employed in EELSMODEL,[8] of the control spectra to the commercial samples was used for quantification of the EELS data. Backgrounds were removed and Fourier-ratio deconvolution routines were applied to all spectra prior to fitting. More specific details of the EELS analysis can be found elsewhere.[6]

2.3 X-ray Photoelectron Spectroscopy (XPS)

Ge-$CeAlO_3$ and bulk CeO_2 were used as controls to generate representative spectra for Ce^{3+} and Ce^{4+}, respectively. The commercial nanoceria materials were analyzed in the form they were sold in, specifically as a powder pressed into copper tape (NPCO sample) or a drop cast suspension onto a silicon wafer (FBC sample).

XP spectra were acquired on an Axis Ultra DLD XPS system from Kratos Analytical (Chestnut Ridge, NY) which was maintained at ultra-high vacuum (UHV) conditions, which was a base pressure of 2 x 10^{-9} torr. Spectra were generated using monochromated Al Kα X-rays to achieve photoemission of core level electrons which were acquired along the sample surface normal with 90% of the photoelectrons collected over a 0.94 mm x 2.25 mm area as determined in previous studies.[9] Due to the insulating nature of the various cerium oxide materials, the surface of the samples were neutralized using low energy electrons to compensate for surface charging. Spectra were acquired at 160 eV pass energy with a step size of 1.0 eV for the wide survey spectra and at 40 eV pass energy with a step size of 0.1 eV for the higher resolution elemental regions.

The acquired spectra were processed using CasaXPS, a commercially available software. All spectra were energy corrected by shifting the C(1s) peak maximum binding energy (BE) to 284.6 eV. All Ce 3d spectra were fit with U2 Tougaard background with the second parameter in the cross-section field adjusted so that the background intersected the noise between the Ce 3d5/2 and Ce 3d3/2 shakedown features for the Ce^{3+} spectra and between the Ce 3d5/2 and Ce 3d3/2 shakeup features for the Ce^{4+} spectra.[10] Semi-quantitative assessment of the distribution of ceria's oxidation states was based on the control sample's spectral lineshapes which were assumed to have 100% of their relative oxidation state. More specific details of the XPS analysis can be found elsewhere.[6]

3. RESULTS

3.1 Orthogonal Analysis of the NPCO Sample

A representative STEM image of the particles from the NPCO sample is shown in Figure 1. The average particle size (mean ± 1SD), as determined from analysis of STEM images, was found to be 20.9 ± 12.5 nm (n = 300).

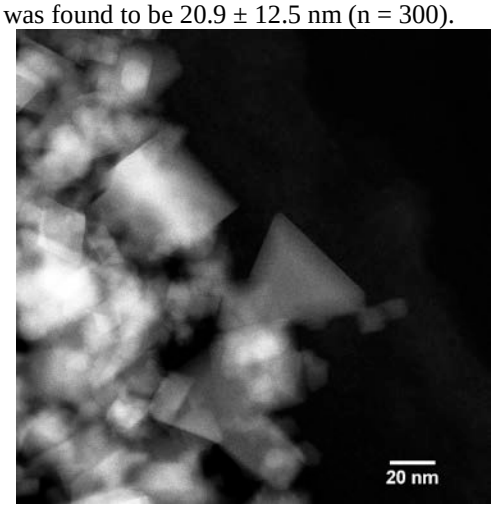


Figure 1. Representative STEM image of the NPCO sample. Scale bar: 20 nm.

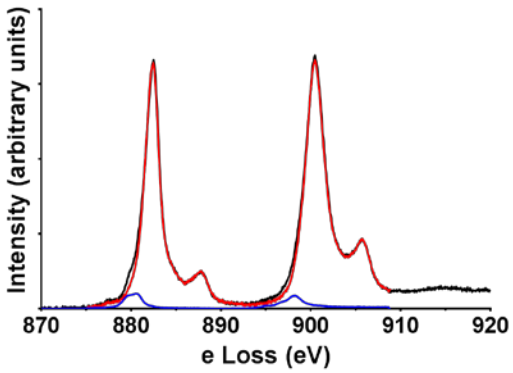


Figure 2. EEL spectra of the NPCO sample. Sample spectrum is shown in black, with contributions to the sample spectra from Ce^{4+} shown in red and contributions from Ce^{3+} shown in blue.

A Ce $M_{4,5}$ EEL spectrum for the NPCO sample is shown in Figure 2. The M edge excitations are due to electron

Gorham, Justin; Hackley, Vincent; Johnston-Peck, Aaron; Maier, Russell; Nelson, Bryant; Sims, Christopher.
"Quantitative Analysis of Oxidation State in Cerium Oxide Nanomaterials."
Paper presented at Technical Proceedings of the 2017 TechConnect World, including the Nanotech 2017 Conference,
Washington, DC. May 14, 2017 - May 17, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

transitions between 3d and 4f states. The 4f state of a Ce^{4+} ion is unoccupied while a Ce^{3+} ion has a single electron. These differences of occupancy are reflected in the electron loss near edge structure (ELNES) of the EEL spectra. The Ce $M_{4,5}$ edges associated with Ce^{3+}, when compared to edges associated with Ce^{4+}, are shifted to lower energies, have different relative peak intensities, as well as different edge shapes (e.g., note the loss of the satellite peaks). In the NPCO sample, the Ce^{4+} contribution dominates most of the spectrum.

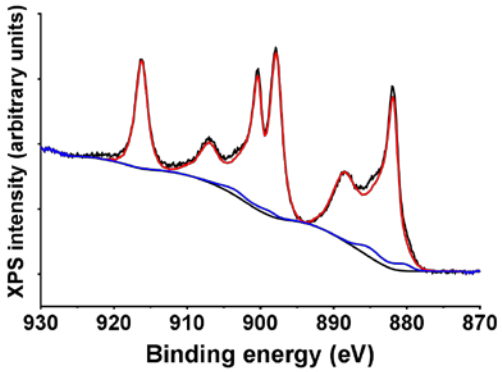


Figure 3. XP spectra of NPCO sample. The sample spectrum is shown in black, with contributions to the sample spectra from Ce^{4+} shown in red and contributions from Ce^{3+} shown in blue.

An XP spectrum of the Ce 3d region of the NPCO sample is shown in Figure 3. Contributions from Ce^{4+}, with the characteristic 6 peaks (two photoelectron peaks, each with a "shake-up" and "shake-down" satellite) dominate the spectrum. Contributions from Ce^{3+}, which consists of 4 peaks (two photoelectron peaks and two peaks from final state effects), comprise very little of the sample spectrum.

3.2 Orthogonal Analysis of the FBC Sample

A representative STEM image of the particles from the FBC sample is shown in Figure 4. The average particle size (mean ± 1SD), as determined from analysis of STEM images, was found to be 4.9 ± 1.3 nm (n = 300).

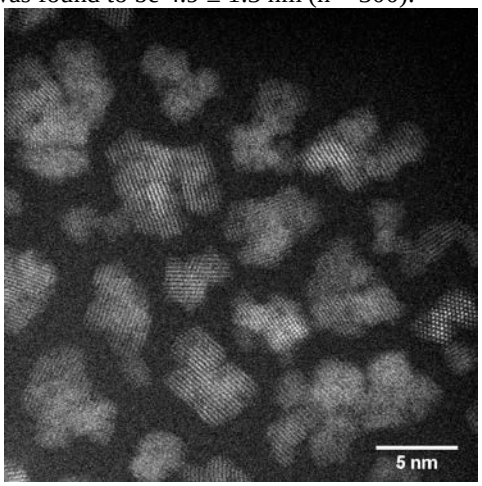


Figure 4. Representative STEM image of the FBC sample. Scale bar = 5 nm.

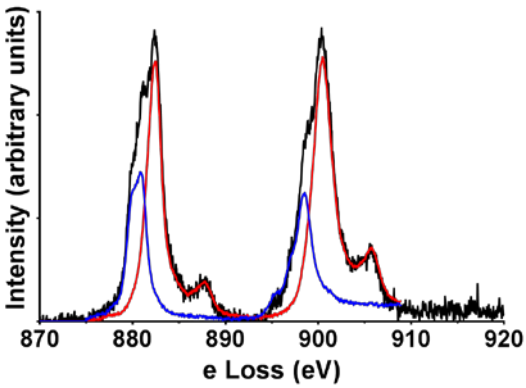


Figure 5. EEL spectra of the FBC sample. Sample spectrum is shown in black, with contributions to the sample spectra from Ce^{4+} shown in red and contributions from Ce^{3+} shown in blue.

A Ce $M_{4,5}$ EEL spectrum for the FBC sample is shown in Figure 5. Ce^{3+} and Ce^{4+} contributions to the spectra are as described in Section 3.1. Note the increased Ce^{3+} contribution in the FBC sample relative to the NPCO sample.

An XP spectrum of the Ce 3d region of the FBC sample is shown in Figure 6. Contributions from Ce^{4+} and Ce^{3+} are as described in Section 3.2. Note the lack of Ce^{3+} contribution to the XP spectrum.

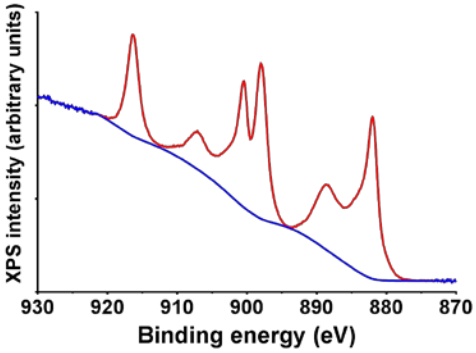


Figure 6. XP spectra of the FBC sample. The sample spectrum is shown in black, with contributions to the sample spectra from Ce^{4+} shown in red and contributions from Ce^{3+} shown in blue.

4. DISCUSSION

Table 1. Ce^{3+} percentage values as determined by each technique on the commercial nanoceria materials.
*: Single calculation based on the fitting of a summation of multiple EEL spectra.

Sample	% Ce^{3+}, EELS*	% Ce^{3+}, XPS
NPCO	5.5	6.4 ± 0.9
FBC	35.2	<0.1

Gorham, Justin; Hackley, Vincent; Johnston-Peck, Aaron; Maier, Russell; Nelson, Bryant; Sims, Christopher.
"Quantitative Analysis of Oxidation State in Cerium Oxide Nanomaterials."
Paper presented at Technical Proceedings of the 2017 TechConnect World, including the Nanotech 2017 Conference,
Washington, DC. May 14, 2017 - May 17, 2017.

This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1239-2

Ce^{3+} percentage values (mean ± 1SD) were determined from each technique for both samples and are listed in Table 1. Based on previous research suggesting that Ce^{3+} content increases with decreasing particle size,[11] the FBC sample is expected to contain far more Ce^{3+} than the NPCO sample. The Ce^{3+} values for the NPCO sample as determined by the analytical techniques are in strong agreement (5.5 % and ~6.5 % for EELS and XPS, respectively). However, values for the FBC sample are quite different between the two techniques (~35 % and <0.1 % for EELS and XPS, respectively). Given the strong agreement of the two techniques for the NPCO sample (and to previous literature trends), this discrepancy is almost surely sample dependent. The FBC sample is comprised of nanoceria suspended in an aliphatic hydrocarbon solvent for its use as a diesel fuel additive. Given the small particle sizes in the FBC sample (and their metal oxide composition), they are expected to have very high surface energies and thus be poorly suspendable in various media. To overcome this, the nanoceria in the FBC are coated by a carbon-based surfactant to keep them suspended. This carbonaceous material was clearly seen in both the XPS C 1s spectrum (data not shown) and interacted with the electron beam during STEM imaging of the FBC sample (data not shown). It is possible that this carbon "contamination" compromised the oxidation state analysis of the FBC sample via XPS. Experiments are underway to determine how to overcome this issue.

5. CONCLUSION

In summary, we demonstrate the use of multiple analytical techniques for the orthogonal characterization of commercially available nanoceria materials using technique-independent controls. While EELS and XPS analyses were in strong agreement for the NPCO sample, the two techniques gave very conflicting results for the FBC sample. It is believed that this discrepancy was an artifact owing to the unique nature of the sample (heavily coated by a carbonaceous material). These results further highlight the importance of thorough characterization of experimental samples using orthogonal approaches due to the potential for differences to arise both from the analytical method chosen *and* the inherent properties of the analyte.

REFERENCES

1. Sun, C. W.; Li, H.; Chen, L. Q., Nanostructured ceria-based materials: synthesis, properties, and applications. *Energy & Environmental Science* **2012,** *5* (9), 8475-8505.
2. Xu, C.; Qu, X., Cerium oxide nanoparticle: a remarkably versatile rare earth nanomaterial for biological applications. *NPG Asia Mater* **2014,** *6*, e90.
3. Baalousha, M.; Ju-Nam, Y.; Cole, P. A.; Hriljac, J. A.; Jones, I. P.; Tyler, C. R.; Stone, V.; Fernandes, T. F.; Jepson, M. A.; Lead, J. R., Characterization of cerium oxide nanoparticles-part 2: nonsize measurements. *Environ Toxicol Chem* **2012,** *31* (5), 994-1003.
4. Spadaro, M. C.; D'Addato, S.; Gasperi, G.; Benedetti, F.; Luches, P.; Grillo, V.; Bertoni, G.; Valeri, S., Morphology, structural properties and reducibility of size-selected CeO2- x nanoparticle films. *Beilstein J Nanotechnol* **2015,** *6*, 60-7.
5. Spadaro, M. C.; Luches, P.; Bertoni, G.; Grillo, V.; Turner, S.; Tendeloo, G. V.; Valeri, S.; D'Addato, S., Influence of defect distribution on the reducibility of CeO 2− x nanoparticles. *Nanotechnology* **2016,** *27* (42), 425705.
6. Sims, C. M.; Maier, R. A.; Johnston-Peck, A. C.; Gorham, J. M.; Hackley, V. A.; Nelson, B. C. *Manuscipt in preparation.*
7. Fu, W. T.; Ijdo, D. J. W., The structure of CeAlO3 by Rietveld refinement of X-ray powder diffraction data. *Journal of Solid State Chemistry* **2004,** *177* (9), 2973-2976.
8. Verbeeck, J.; Van Aert, S., Model based quantification of EELS spectra. *Ultramicroscopy* **2004,** *101* (2–4), 207-224.
9. Barron, S. C.; Gorham, J. M.; Patel, M. P.; Green, M. L., High-Throughput Measurements of Thermochromic Behavior in V1−xNbxO2 Combinatorial Thin Film Libraries. *ACS Combinatorial Science* **2014,** *16* (10), 526-534.
10. Salvi, A. M.; Decker, F.; Varsano, F.; Speranza, G., Use of XPS for the study of cerium–vanadium (electrochromic) mixed oxides. *Surface and Interface Analysis* **2001,** *31* (4), 255-264.
11. Deshpande, S.; Patil, S.; Kuchibhatla, S. V. N. T.; Seal, S., Size dependency variation in lattice parameter and valency states in nanocrystalline cerium oxide. *Applied Physics Letters* **2005,** *87* (13), 133113.

Gorham, Justin; Hackley, Vincent; Johnston-Peck, Aaron; Maier, Russell; Nelson, Bryant; Sims, Christopher.
"Quantitative Analysis of Oxidation State in Cerium Oxide Nanomaterials."
Paper presented at Technical Proceedings of the 2017 TechConnect World, including the Nanotech 2017 Conference, Washington, DC. May 14, 2017 - May 17, 2017.