Volume 104, Number 4, July–August 1999
Journal of Research of the National Institute of Standards and Technology

[J. Res. Natl. Inst. Stand. Technol. **104**, 401 (1999)]

Conference Report

SECOND ADVANCED ENCRYPTION STANDARD CANDIDATE CONFERENCE Rome, Italy March 22-23, 1999

Report prepared by

Morris Dworkin

Computer Security Division,
Information Technology Laboratory,
National Institute of Standards and Technology,
Gaithersburg, MD 20899-0001

Available online: http://www.nist.gov/jres

1. Introduction

On March 22-23, 1999, nearly 200 members of the global cryptographic research community gathered in Rome, Italy for the Second Advanced Encryption Standard Candidate Conference (AES2). This report summarizes the conference presentations and accompanying discussions. AES2 was the second of three conferences sponsored by the National Institute of Standards and Technology (NIST) in its effort to develop a new encryption standard for the U.S. Government. There are 15 candidate algorithms in Round 1 of the analysis period, out of which NIST will select about five finalists in the summer of 1999 for further evaluation in Round 2; the main purpose of the conference was to advise NIST in the selection of these finalists.

The goal of this development process is to produce a Federal Information Processing Standard (FIPS) for an Advanced Encryption Standard (AES) specifying an Advanced Encryption Algorithm(s) (AEA), for use by the U.S. Government and, on a voluntary basis, by the private sector. According to NIST's formal call for algorithms, published on September 12, 1997:

> It is intended that the AES will specify an unclassified, publicly disclosed encryption algorithm available royalty-free worldwide that is capable of protecting sensitive government information well into the next century. [1]

NIST requires the AES to be a symmetric key block cipher that (at a minimum) supports a block size of 128 bits and key sizes of 128 bits, 192 bits, and 256 bits. The AES is expected to succeed the Data Encryption Standard (DES), whose 56 bit key is becoming vulnerable to exhaustive search.

NIST maintains an AES homepage at http://www.nist.gov/aes; see also [2] for a thorough discussion of the AES development process and a summary of the First AES Candidate Conference, including brief technical descriptions of the 15 candidate algorithms.

2. Welcome and Overview

William Wolfowicz, of the Fondazione Ugo Bordoni, and the European coordinator of the conference, briefly welcomed the participants to the meeting and to Rome. Miles Smid, the Acting Chief of the Computer Security Division of NIST's Information Technology Laboratory, spoke at greater length to open the proceedings. He began by expressing his satisfaction at the turnout (180 registered participants representing at least 23 countries) and the number of papers to be presented (21). He thanked the program committee for their work, and he said that all of the papers submitted to the con ference would be available on the AES homepage.

Smid then outlined the program. There were three general conference goals: to present Round 1 analysis of the AES candidates, to discuss relevant issues, and, especially, to provide NIST with a clearer understanding of which candidate algorithms should qualify for Round 2 and which should not. The three main issues to be addressed at the conference were security, efficiency, and flexibility; these were the main factors that NIST originally identified for evaluating the algorithms. In the area of security, there were talks on cryptanalysis, power analysis and related attacks, and the concept of "minimal secure rounds." Some of the cryptanalytic attacks were already known, but they had not yet been presented formally at a conference. In the area of efficiency, there were several surveys comparing the candidates on various 8 bit, 32 bit, and 64 bit platforms. Two other issues that probably would be addressed were intellectual property and the possibility of selecting multiple winners for the AES.

The conference was organized into seven sessions. On the first day, Sessions 1 and 2 were devoted to surveys; Sessions 3 and 4 covered smart card implementations and related attacks. In addition, Mr. Smid invited the attendees to submit proposals for short talks to give in the evening "rump session." On the second day, cryptanalytic attacks were slated for Session 5, and algorithm observations for Session 6. Session 7 was devoted to algorithm submitter responses and to a discussion of issues, including audience questions.

3. Surveys (I)

The chair of Session 1, Tom Berson of Anagram Labs, offered the audience some advice in listening to the upcoming presentations of survey results. He said that the authors would propose some criteria, perhaps explain their relevance, present a table of measurements against the criteria, and perhaps draw conclusions based on the data. He advised the audience to view the talks with an open mind but also with a healthy skepticism, bearing in mind the people who created the data, any agenda they might have, the relevance of their criteria, and any actions they advocated.

The first speaker of the session was James Foti, a mathematician with NIST's Security Technology Group. He presented the results of the efficiency testing that NIST, in its formal call for algorithms, had indicated it would perform for Round 1. NIST had specified the following reference configuration: a Pentium Pro, 200 MHz, with 64 MB of RAM, running Windows95, using the ANSI C compiler in the Borland C++ Development Suite 5.0. Foti emphasized that these timings would be only part of the information that NIST would consider in choosing the finalists, and NIST did not necessarily expect its results to be the fastest possible on a 32 bit processor. Another caveat was that NIST used the submitters' C code, which probably varied in the degree of optimization.

Foti explained the measurement techniques that NIST used in the timing program and the clock cycle program for its ANSI C testing. Then he presented timings of implementations on the reference configuration for encryption, decryption, and key setups. Only 128 bit keys were considered; larger key sizes would be tested in Round 2. He then compared these results with those of Brian Gladman and those of the Twofish team (which were scheduled to be presented as part of the same panel). Whereas NIST used the optimized code required in the AES submissions, Gladman wrote his own code, and the Twofish team used several sources. Although there were some discrepancies, which he discussed, the bottom line was that the three surveys shared the same set of five fastest algorithms: CRYPTON, MARS, RC6(tm), Rijndael, and Twofish. Also, among the five slowest algorithms in each of the three surveys were DEAL, LOKI97, and MAGENTA.

Foti also indicated several other combinations of processors, operating systems, and compilers on which NIST had conducted tests of the submitted ANSI C code. Instead of presenting individual data on these, he presented averages of the results obtained from different compilers on two of the platforms. He also presented some results from NIST's ongoing testing of AES Java(tm) implementations, including static and dynamic memory as well as speed. The fastest algorithms there, in order, were Rijndael, MARS, CRYPTON, LOKI97, CAST256, and Twofish.

Foti concluded by noting that similar groupings existed among different implementations of the algorithms on 32 bit processors; NIST would also need to look at performance figures on 8 bit platforms and on 64 bit processors, some of which would be presented later at the conference. For Round 2, NIST planned to test the larger key sizes, and to run the C code on 64 bit processors using compilers that generate 64 bit applications. In addition, NIST is considering the possibility of testing assembly language implementations.

Brian Gladman was unable to attend AES2, so instead Berson read excerpts from his paper. Gladman had coded and implemented each of the candidates; the paper was intended to share his experience and to provide fair and accurate comparisons, not only in performance results, but also in the ease of implementation. For example, he discussed the form and character of the specifications, the degree of guidance given for implementation options and optimization opportunities, and the attention to byte order "endianness." Berson recommended the paper.

Bruce Schneier of Counterpane Systems spoke next, presenting the joint work of the Twofish team comparing the performance of the candidates in several settings: Pentium Pro/II, Pentium, DEC Alpha, 8 bit smart cards, and hardware. Before presenting their results in those areas, Schneier discussed the effect of larger key sizes on the performance of the candidates. He also offered two opinions on how to best compare and evaluate performance. First, he claimed that the AES would have to perform on all types of processor architectures, and that performance was more important on the "low end." Second, he advocated comparisons in assembly language over C and Java, because applications where speed was important would be coded in assembly language .

The first area of comparison was on 32 bit processors, specifically, the Pentium Pro/II and the Pentium. Schneier presented encryption timings and estimates, noting that the performance varied greatly, and that the performance of some algorithms depended heavily on the CPU. In particular, MARS and RC6 performed relatively better on the Pentium Pro/II, whose CPU supports fast 32 bit multiplication and variable bit rotations. For seven of the algorithms, the analysis was extended to include key setup; Rijndael and CRYPTON were the fastest algorithms for small blocks, although all of the speeds settled down pretty quickly. Along the same lines, Schneier compared the suitability of the candidates as hash functions. He presented results based on Biham's "minimal secure variants": Twofish and Rijndael became the fastest algorithms, although he did not necessarily endorse Biham's idea.

In the area of 64 bit CPUs, the Twofish team estimated the performance of most of the algorithms on the DEC Alpha; the fastest algorithms, in order, were DFC, Rijndael, Twofish, and HPC. In the area of smart cards, the Twofish team concentrated on 8 bit processors. Schneier cautioned against comparing numbers in the various papers because the underlying assumptions varied. He asserted that memory usage was an essential consideration: DFC, E2, MARS, and RC6 could not realistically fit on small smart cards; FROG could not fit on any smart card. In the area of hardware, the Twofish team did not try to count gates; instead, they concentrated on context switching speeds. They cited CRYPTON as the most hardware-friendly algorithm; Rijndael and Twofish were also efficient in hardware. Schneier then summarized the findings for each individual algorithm, and invited the audience to draw its own conclusions.

The last speaker of Session 1 was Alan Folmsbee of Sun Microsystems, Inc. There were three main components to his analysis. First, he presented his "fracstel number," a measure he invented to try to normalize the concept of a round, and applied it to each candidate. Second, for each candidate, he determined the minimal number of rounds at which the avalanche was nearly ideal, in his estimation, and then measured the "excess avalanche." Third, he presented some Java timings obtained from submitters' Java code run on a 200 MHz UltraSPARC™, along with ROM measurements and RAM estimates; the six fastest candidates were MARS, RC6, E2, Serpent, HPC, and CRYPTON. He concluded with a personal recommendation of five finalists based on a weighted average of the rankings of the candidates in his various criteria. They were, in order, RC6, MARS, SAFER+, Serpent, and CRYPTON.

4. Surveys (II)

Serge Vaudenay of the Ecole Normale Supérieure-Centre National pour la Recherche Scientifique, the first speaker of Session 2. presented the DFC team's report on the candidates. He began with comments on the use of ANSI C in the reference configuration. Some usable instructions were restricted; the standard implementations of seven candidates did not turn out to be portable to SPARC™ or Alpha machines; and if a certain conversion was coded carefully, then the candidate unfairly incurred a performance penalty. He agreed with Schneier that it was better to compare the candidates in assembly language than in C. He also claimed that the Pentium Pro was outdated technology and that it would be better to compare the candidates on RISC 64 bit microprocessors like the Alpha.

Vaudenay then presented the timing results of two colleagues: Granboulan's work on the Alpha and Noilhan's Java implementations on the UltraSparc-I™. On the Alpha, DFC was clearly the fastest algorithm, with HPC second, followed by Rijndael and Twofish, which were slightly faster than CRYPTON and MARS. For Java coding, RC6, Rijndael, MARS, HPC and Serpent were the fastest.

In the remainder of the talk, Vaudenay commented on various algorithms against the following criteria: speed, origin of S-boxes, simplicity, portability, and the underlying research. He touted the theoretical design of DFC, which supplements a conservative design with a decorrelation module. He touched on the security of three algorithms: a class of weak keys in CRYPTON—also noticed by Johan Borst; attacks on DEAL; and a preliminary, theoretical, statistical attack on reduced-round RC6. The four finalists he recommended were DFC, MARS, RC6, and Serpent.

Craig Clapp of PictureTel Corporation, the second speaker of the session, considered the parallelism of seven of the candidates at the instruction level. There

were two parts to his work: a theoretical analysis of the "critical path," and a practical simulation of the algorithms' performance on a family of machines with a specified instruction set and from one to eight "execution units." The results of the simulation matched the critical path analysis well. For up to three execution units, RC6 was the fastest algorithm; MARS, Rijndael, and Twofish had virtually identical performance. At more than four execution units, Rijndael was the fastest, followed, in order, by CRYPTON, RC6 and Twofish, MARS, E2, and Serpent. Clapp concluded that CRYPTON and Rijndael seemed to be the candidates best able to benefit from increasing instruction-level parallelism.

Eli Biham of Technion spoke next on his method of normalizing the speed comparisons of the algorithms for security. His underlying observation was that NIST had not specified a relation between strength and speed, and so there were varying security margins among the candidates. Serpent, for example, with 32 rounds, had a large margin of security, since the designers believed 16 rounds to be secure; other algorithms had smaller margins, adding just a few rounds to the minimal number at which the algorithm was believed secure from attacks stronger than exhaustive search. He proposed a "fair speed/security" comparison, in which the algorithms would be evaluated at two passes more than these minimal secure variants. He claimed that under this model, the fastest candidates, in order, were Twofish, Serpent, MARS, Rijndael, and CRYPTON.

Here are some of the questions that the audience posed to Biham. Was it fair to add two extra passes since that could represent different margins of security for different algorithms? Biham acknowledged the difficulty, noting that it was impossible to add fractional rounds, and inviting others to vary his scheme. How did he arrive at his estimate that there was an attack on CAST-256 reduced to 32 rounds? Biham thought there was something to that effect in the CAST-256 submission paper, but he admitted that he could be mistaken; however, he did personally know of attacks that break CAST-256 with more than 20 rounds. Was he thinking of publishing them? Yes, he might, now that he knew that they were the best results. Why did he not perform the adjustments on the best available timings instead of timings that seemed to favor Serpent? Biham responded that he had merely used timings from his own computer; he invited others to base a comparison on Gladman's timings. Did Biham advocate that the number of rounds should be a changeable parameter? Not necessarily, but as it stood, it was like comparing apples with oranges: some algorithms were designed more for speed and others more for security.

To close the session, Foti spoke again to report on NIST's Round 1 randomness testing. In addition to performing its own statistical tests, NIST used the Crypt-XB, and DIEHARD statistical packages. He explained how the tests were conducted and presented the empirical results. As expected, output of all of the algorithms looked random; no statistically significant results were discovered. In the future, NIST planned to conduct analysis on the larger key sizes and possibly on reduced round versions.

5. Smart Cards: Implementations and Related Attacks

Session 3 consisted of two partial surveys of smart card implementations. François Koeune spoke first about the work of the Cryptography Group at the Universit Catholique de Louvain. They performed implementations of E2, RC6, Rijndael, and Twofish on emulators of two different smart cards: the Intel 8051 and the ARM. The former was a basic, low-cost, 8 bit smart card, and the latter was sophisticated and advanced, with a 32 bit processor. Koeune explained some of the implementation decisions; for example, they gave RAM usage priority over speed. E2 performed the slowest and used the most RAM on both smart cards, more than was available on the 8051. Rijndael used the least RAM on both smart cards and also performed the fastest. Twofish was the second fastest on the 8051, and required relatively little RAM on both smart cards, while RC6 was the second fastest on the ARM. Work on MARS and Serpent was in progress.

Geoffrey Keating presented a survey of several candidates on the Motorola 6805 series 8 bit architecture, allowing a maximum of 120 bytes of RAM, which he considered to be generous, and 1024 bytes of ROM. He quoted published results for Twofish, and he implemented "constant-time" simulations of five other candidates himself; he also looked at E2 and CAST256. He presented and discussed his findings [3], updating those in the published proceedings. Rijndael was the fastest, followed, in order, by Twofish, CRYPTON, Serpent, RC6, and MARS. MARS exceeded available ROM significantly, as would CAST-256; RC6 exceeded RAM limits.

Session 4 consisted of three talks on implementation attacks. Before presenting the first paper of the session, Adi Shamir of the Weizmann Institute of Science addressed one of the questions that arose after Biham's talk: should the number of rounds be changeable? Shamir proposed that NIST postpone any changes in the number of rounds of the algorithms until Round 2; then, in consultation with the submitters of the finalists,

specify a different number of rounds for each key size. The guideline would be to use a couple of rounds more than the minimal secure rounds for 128 bit keys, to use twice as many rounds for the 256 bit keys, and to use an intermediate number of rounds for 192 bit keys.

Shamir then presented his and Biham's paper on power analysis. He stressed the practical importance of implementation attacks, since the eventual AES winner was likely to be very secure against classical attacks. The general idea, due to Kocher [4], was to measure and analyze the power consumption of a smart card to reveal the key. In Shamir's variant, the power consumed by writing subkey bytes into RAM gives a measure of their Hamming weights. In DES, for example, such measurements would yield 96 noisy equations in the 56 bits of the user key, which in principle would be more than sufficient to calculate it. This variant of power analysis was important because it focused directly on the key schedule of a cipher, independent of the plaintext, ciphertext, protocol, and implementation details. Shamir discussed examples of potentially dangerous instructions in the key scheduling of the AES candidates.

In the question-and-answer period after the talk, Shamir was asked about the resistance of the AES candidates to the attack; he asserted that a few were vulnerable, but hesitated to assert that any were not vulnerable. Schneier commented that implementation attacks were best resisted at the level of hardware or protocols; Shamir agreed, although the hardware defenses he had seen were problematic too.

The next speaker, Joan Daemen of Proton World, presented his and Rijmen's survey of the candidates' resistance to timing attacks, simple power analysis, and differential power analysis. He discussed the impact of various operations in resisting these attacks: storing and loading registers, rotations and shifts, bitwise Boolean operations, and arithmetic operations. He also discussed possible countermeasures. In his conclusion, he classified the candidates in three ways. CAST-256, DFC, E2, HPC, MARS, and RC6 were problematic because they used multiplication and/or variable rotations. FROG, LOKI97, SAFER+, and Twofish were doubtful because they used addition and/or subtraction. CRYPTON, DEAL, MAGENTA, Rijndael, and Serpent were favorable because they did not use arithmetic operations or variable rotations.

Pankaj Rohatgi, the last speaker of the session, presented the work of a team of cryptographers, statisticians, hardware experts, and smart card experts at IBM's T.J. Watson Research Center. The main conclusion was that NIST's flexibility criterion for "secure and efficient" implementations was very hard to achieve on smart cards because of an inherent susceptibility to physical attacks like Kocher's differential power analysis. For example, the team was able to obtain the whitening subkeys of Twofish using only 50 power samples from a straightforward implementation on a ST16 smart card. Rohatgi estimated that, under their attack model, all the candidates were vulnerable, although some more than others. In particular, FROG and HPC would be the least easy to attack, and that CAST-256, DFC, E2, MARS, and RC6 would be less easy to attack than the remaining eight algorithms. He presented some possible countermeasures and their associated overhead. He recommended that NIST revisit one of their selection criteria: smart card performance should either be dropped, or it should only be compared on power analysis resistant implementations, which would probably require advanced smart card platforms.

6. Rump Session

Several attendees gave short talks for the rump session, which was held on the first evening. Smid spoke first on behalf of Don Johnson of Certicom, who could not attend the conference. Johnson contended that the AES ought to specify multiple algorithms in order to ensure its future resiliency: if one algorithm was found to have a fatal flaw, then another one would be readily available.

Ross Anderson of Cambridge University spoke about the security of smart card implementations against both invasive and non-invasive attacks. Although hardware protection might be possible, the cryptographic research community was only seeing the tip of the iceberg. He proposed that the AES finalists should be implemented in hardware in order to evaluate their resistance to such attacks.

Orr Dunkelman of Technion spoke about the security of Serpent-p and Serpent-p-ns (two variants of Serpent with a weaker linear transformation) against linear cryptanalysis, differential cryptanalysis, and impossible differential cryptanalysis.

Ian Harvey of nCipher Corporation Limited presented a class of implementation attack applied to DFC.

Kazumaro Aoki of Nippon Telegraph and Telephone (NTT) Laboratories spoke about the performance of E2. He disagreed with NIST's Java test data, asserting that the impact of the NIST API on the encryption performance in NIST's timings was not negligible. He presented new optimization methods for E2, and he presented a performance comparison on a Pentium II in which the five fastest candidates were RC6, Twofish, Rijndael, E2, and MARS. He urged the use of the latest results.

Shamir addressed a question that arose during his earlier talk, explaining how the timing of implementa tion details could impact security.

Schneier commented on Biham's idea for minimal secure round variants. He agreed that the notion of a "conservativeness" measure was a good one, but both the strength of the valid attacks and the measures of safety needed to be carefully defined.

Shiho Moriai of NTT Laboratories presented a measure of the randomness of three structures in the candidates: Feistel structure, MARS-like, and CAST-256-like.

Johan Borst of K.U. Leuven spoke on weak keys of CRYPTON, the subject of an official comment that he submitted as an official comment in August 1998; he also acknowledged later contributions by Vaudenay, Wagner, and their teams. These results made CRYPTON unsuitable for use in hashing.

Doug Whiting of Hi/fn, Inc. presented performance comparisons of RC6, Rijndael, Serpent, and Twofish on Merced.

Niels Ferguson of Counterpane Systems recommended an emergency mode for AES, in which the number of rounds would double, instead of choosing multiple algorithms.

Takeshi Shimoyama of FUJITSU Labs Ltd. spoke about the security of Serpent's S-boxes against higher order differential attacks, disputing the claim in Serpent's documentation that all of its S-boxes have nonlinear order 3.

David Wagner of the University of California Berkeley explained how HPC's lossy key-expansion made it easy to find equivalent keys, so that HPC was unsuitable for use in hashing.

Ron Rivest of the MIT Laboratory for Computer Science presented a possible alternative key schedule for RC6 that could be calculated forwards and backwards on the fly. He claimed that RC6 was modular in that the key schedule could be considered separately from encryption.

Chae Hoon Lim of Future Systems, Inc. presented a hardware architecture design of CRYPTON version 1.0, a revision of the original submission. He discussed the design decisions and results.

Schneier spoke against the idea of multiple algorithms in the AES, arguing that it would mean higher costs, especially in hardware, and, in some respects, less security. He would even prefer that Twofish not be chosen at all, rather than having it included in a suite of multiple algorithms.

Gary Graunke of Intel spoke on critical path opcode analysis, comparing ideal AES times to observed times.

Carl Ellison of Intel presented a humorous new metric for comparing the algorithms.

7. Announcement of the Third AES Conference

Before Session 5, Smid thanked the organizers of the Fast Software Encryption Workshop 1999 (FSE6) for allowing NIST to hold the AES conference during the same week at the same venue. He announced that the next AES conference would be coordinated with FSE7 in New York City in April 2000.

8. Cryptanalysis

Session 5 was devoted to cryptanalysis of the candidates. Sean Murphy of the University of London presented his and Mirza's paper on two properties of the key schedule of Twofish, focusing on the 128 bit key case. First, not all pre-whitening subkeys could occur. Murphy said that the Twofish team had further results: the distribution of subkeys was slightly less uniform than he predicted in his paper [5]. Second, Twofish could be considered as a collection of 2^{64} versions of "reduced Twofish," in which the round functions are fixed by the selection of one of the possible pairs of key-dependent S-boxes. Because the subkey generation of reduced Twofish was unbalanced, guessing the S-boxes would yield an attacker a slight amount of information, contrary to a claim in the Twofish submission. This raised the possibility that the imbalance could be exploited for some key classes, which would constitute a divide-and-conquer attack on those classes.

John Kelsey of Counterpane Systems then presented joint work with Schneier and Wagner on weaknesses in the large key size versions of SAFER+. The underlying weakness was the poor key diffusion in the key schedule; in other words, it took several rounds before changing certain key bits would affect the cipher. He described two attacks of academic interest on the 256 bit key version. The first was a meet-in-the-middle attack requiring 2^{240} work, 12×2^{24} bytes of memory, and 3 texts. Besides the poor diffusion, this attack also exploited a property of the linear transformation in the round function. The second was a related-key attack requiring 2^{200} work and 3×2^{32} chosen texts under each of two related keys. Neither attack was practical; nevertheless, Kelsey suggested improvements in the key schedule for the larger key sizes that eliminated the poor key diffusion.

Vincent Rijmen of the University of Bergen presented joint work with Knudsen on the security of LOKI97. He explained how certain weaknesses could be exploited in differential and linear attacks. There were several two-round iterative differentials based on the noninvertibility of the S-boxes and the invariance, under modular addition of subkeys, of input pairs that

differed only in the most significant bit. The probability of the best 15-round characteristic was 2^{-56}, and this would be compounded by resynchronization; 2^{56} chosen plaintexts should suffice for an attack. The linear attack exploited the correlation of the least significant bits of the inputs and outputs under modular addition, as well as the imbalance in the S-boxes used in the second layer of the round function. The probabilities of the two best 15-round linear approximations that they found were 2^{-22} and 2^{-29}, for 2^{-14} and 2^{-7} of the keys, respectively.

Wagner presented joint work with Ferguson and Schneier on a weak class of keys in FROG. There were two underlying observations. First, FROG's S-boxes and internal wiring depended on the key, so the quality of diffusion did as well. Second, the diffusion was much worse in the reverse direction than in the forward direction. They exploited these properties in a differential attack using 2^{36} chosen ciphertexts on 2^{-56} of the keyspace. Wagner acknowledged an error in their paper: they had claimed that it would be easy to recover the entire S-box when, in fact, only about half of the S-box yielded easily; however, they still suspected that with more work it would probably be possible to recover the full S-box. He also discussed a dual linear attack. In response to a question from Dianelos Georgoudis, the submitter of FROG, Wagner said that there was probably a quick fix to eliminate this particular weak class, but he would not be confident that there were no other such classes.

In the last talk of the session, Biham presented results on MAGENTA that he had written at the first AES conference with several other attendees. They mounted a simple attack exploiting the symmetry of the key schedule. For the 128 bit key version, the attack required 2^{64} chosen plaintexts and 2^{64} steps of analysis; alternatively, it could be converted to an attack with 2^{33} known plaintexts and 2^{97} steps of analysis. The same attacks on the larger key sizes resulted in the same reduction of complexity over exhaustive search.

9. Algorithm Observations

Session 6 was devoted to observations on individual algorithms. Jacques Stern of the Ecole Normale Supérieure advocated DFC on behalf of its design team. He highlighted the "provable security" features, which protected against certain delimited attacks, under the assumption that the subkeys behaved randomly. He also assumed that the conservative design for the confusion permutation protected against other attacks. He backed up both of these assumptions with two specific security "challenges." He also highlighted DFC's performance on 64 bit architectures, which he called "tomorrow's architecture," and on which DFC was the fastest candidate. He explained how candidate comparisons that used the Pentium Pro or ANSI-C unfairly penalized DFC. He cited other implementations to make the case that DFC was not in the trailing group of candidates for speed. He addressed Coppersmith's weak keys: they only occurred with probability 2^{-128}, and a slight modification in the key schedule would fix the problem.

Kazukuni Kobara of IIS Tokyo University presented a very technical paper, "Pseudorandomness and Maximum Average of Differential Probability of Block Ciphers with SPN-Structures like E2," on behalf of Makoto Sugita. The conclusion was that the linear transformation in E2 provided good pseudorandomness and good immunity against differential attacks.

James Massey of Cylink Corporation spoke next on the linear transformation of SAFER+. He explained how it provided diffusion that was optimal among a certain class of transformations that lent themselves to fast implementations, called "multi-dimensional 2-point transform diffusers." He also reported that both software and hardware implementations of SAFER+ had been improved significantly; details were available at the SAFER+ Forum at the AES homepage.

Scott Contini of RSA Laboratories presented joint work with Yin on the operation of data-dependent rotations, which were used in MARS and RC6. They conducted an extended analysis of how input differences in the rotation amount affected data-dependent rotations. Their results confirmed the intuition that such differences provide a fast avalanche of change. He concluded that MARS and RC6 appeared to resist differential cryptanalysis.

Whiting spoke next on behalf of the Twofish team with new findings on the security and efficiency of their algorithm. He began by briefly addressing the two security issues that Murphy had raised in his talk that morning, referring the audience to the paper on their website for a thorough discussion [5]. First, they calculated that the entropy of the whitening subkeys was 117 bits out of 128 bits, which was even less than Murphy conjectured; however, that was not significant, because only 64 bits were needed to mask the input to the S-boxes in the first round, as in RC6, for example. Second, they empirically estimated from smaller cases that, for a given choice of S-boxes, the entropy of the subkeys was 63.2 bits out of 64 bits. Whiting claimed that this was also not a significant security concern; DES, for example, lost 4 bits of entropy per round in an analogous situation.

Whiting then summarized the results in their conference paper. They had empirically verified some uniqueness properties of the Twofish keys; for example, no two distinct user keys produce an identical sequence of

subkeys. He also explained the results of their effort to derive an upper bound on the probability of a differential characteristic. But most of his talk focused on implementations of Twofish in various settings. There were speed improvements in the fastest assembly language implementations of Twofish. Whiting also presented results that illustrated the performance flexibility for which Twofish was designed: not only encryption speed versus key setup speed, but also RAM versus key setup speed. There were similar tradeoffs in hardware, including a new hardware implementation that used only 8000 gates.

10. Algorithm Submitter Rebuttals and Discussion

Smid moderated Session 7, the last session of the conference. The algorithm submitters sat as a panel; every algorithm except HPC and LOKI97 was represented. Individual submitters had an opportunity to speak for a few minutes, followed by an update from Smid on the intellectual property situation of the algorithms. Then the attendees participated in an open discussion of various issues.

Six submitters delivered statements. Massey presented a revised, "unified" key schedule for SAFER+ that addressed the weaknesses in the two larger key sizes, while reducing to the original key schedule in the 128 bit case.

Klaus Huber of Deutsche Telekom AG spoke on behalf of MAGENTA. He asserted that criticisms of MAGENTA were exaggerated and sometimes incorrect; for example he disputed the assertion in the survey of the candidates by the Twofish team that it would be hard to implement MAGENTA in hardware faster than 180 MB/s. He advocated several aspects of MAGENTA. The key schedule weakness could be easily eliminated. MAGENTA was one of the top candidates in memory usage, resistance to implementation attacks, and key setup. It was very fast in hardware, and its software performance could be improved with the use of 16 bit tables. Last, its design was clear and compact. In response to a question from the audience, Huber said he was in the process of selecting one of several possibilities for a new key schedule.

Carlisle Adams of Entrust Technologies spoke about CAST-256. He asserted that it appeared to be secure at 48 rounds; the best attacks of which he was aware were on 16 and 20 round versions. He said that some people had suggested that since there were not many results in the area of the first selection criterion, security, the next criteria ought to be comparisons of performance, code size, memory requirements, etc. Adams disagreed: the second criterion also ought to be security, in particular, the security history of any predecessors of the candidates. He traced the 6 years of history of the three iterations of CAST, none of which had been broken. CAST-256 used the same round function that had already been well studied; the two modifications were the key schedule and the extended Feistel framework. Adams argued that it was much more manageable to evaluate the security of those two modifications than to examine every detail of a brand new cipher. He also discussed the issue of performance. CAST-256 fell quite comfortably in the middle of the pack, only 2 to 5 times worse than the fastest candidates, which would not be noticeable in many common environments. He suggested that in the AES process, solid performance in every environment was desirable.

Lim commented on CRYPTON. First, he pointed out that it featured a two step key generation procedure to facilitate low-level implementations: expansion into an extended key, and then the generation of round subkeys. In smart card implementations, the expansion could be performed just once, and the extended key could be stored, making irrelevant a certain power analysis attack. He asserted that several of the survey papers inflated the key setup time for CRYPTON; in fact, it was faster than one encryption in almost every architecture. He urged the use of his figures for comparison. Last, he pointed out that the motivation for Vaudenay's differential and linear attacks had already been considered in the original CRYPTON documentation.

A submitter of RC6, Matt Robshaw of RSA Laboratories, spoke next. He wanted to raise the issue of cross-compiler timings; he hoped that there would be a discussion on how to compare the candidates fairly across different compilers. He questioned NIST's Java timings for RC6, and also E2; those of Folmsbee and Vaudenay were more in line with their expectations. He also presented a ranking of the minimal secure variants of the algorithms that was based on NIST's timings, as quoted by Schneier; unlike Biham's ranking, the "usual suspects" came out on top.

A submitter of E2, Kazuo Ohta of NTT, also questioned NIST's Java timings; he referred the audience to a conference handout for NTT's results.

Smid then updated the audience on the intellectual property (IP) statements of the candidates. The IP goal was for AES to be available royalty-free worldwide. Although the submitters had agreed to give up their own IP rights if their algorithm was chosen for the AES, NIST was concerned that submitters whose algorithms were not chosen might claim that a winner infringed their IP. NIST informally polled the submitters on this question; the submitters that agreed to waive their IP

rights on the winner without qualification were CAST-256, CRYPTON, DEAL, FROG, LOKI97, Rijndael, Serpent, and Twofish. MAGENTA had not responded until the conference, when Huber said he could not speak for the position of Deutsche Telekom, which held patents on the fast Fourier transform. Smid presented slides with the responses of the other candidates, who had appeared to qualify their responses in some way, for example, by seeking recognition for their ideas.

Smid invited the submitters to clarify their responses, and opened the floor for discussion. Someone pointed out that IBM's response appeared to waive the exercise of its MARS patent but not any of IBM's other patents. An attendee from IBM explained that they were concerned that the wording of NIST's poll question could be interpreted to cover *any* actions of the users of the AES, not just the use of the AES itself; Massey said that Cylink Corporation shared this concern. Other attendees raised the concern of possible IP claims from non-submitters. Smid responded that he was not sure if it was possible to avoid them, beyond trying to conduct a good patent search. Nevertheless, he repeated his request from the first AES conference: any IP claims on any of the candidate algorithms should be brought to NIST's attention.

A few other issues were discussed. There was not a clear consensus on whether the AES should specify multiple algorithms. The original idea in Johnson's paper was that diversity increased security; Smid pointed out that multiple algorithms also would be a kind of insurance against IP disputes. Someone suggested that, in light of the threat of implementation attacks, it might be appropriate to have performance diversity in the standard: one algorithm for smart card environments and another for protected environments. This idea met with some objection, however: for example, it might unfairly penalize submitters who had taken the effort to design algorithms with the requisite flexibility. Another attendee recommended that, whether or not multiple algorithms were selected, AES protocols should support an eventual change from 128 bit keys to 192 bit keys; moreover, constant-time implementations across the key sizes would be desirable to minimize the impact of that change. This sparked the observation that, absent a performance penalty for larger key sizes, there was little incentive to use smaller key sizes.

The observation that requiring multiple algorithms would increase costs—although not so much in modern toolkits, it was later pointed out—sparked a discussion of whether AES needed to fit at all on low end smart cards. On the one hand, if AES only fit on more sophisticated smart cards, then costs would rise significantly. On the other hand, if the information being protected was valuable enough to require use of the AES, then perhaps the extra cost was appropriate, in which case it would be unfortunate to skew the AES selection towards performance in limited environments.

Smid asked for comments on the usefulness of "provable security" and "minimal secure rounds." Schneier offered the only response to the former: it was useful and helpful in analysis, but only as good as the underlying assumptions and model. He likened it to the analysis that many teams had provided against certain types of attacks; it increased confidence, but he would not choose a cipher based on that factor alone.

The question of "minimal secure rounds" attracted more discussion. One attendee believed that the margin of safety was essentially independent of the algorithm design; therefore, NIST, with input from the cryptographic community, ought to give guidance for the margin of safety, to avoid losing otherwise good algorithms. However, it was pointed out that determining the margin of safety for each algorithm was still a problem, especially since the candidates had weathered different levels of analysis. It was suggested that NIST allow round variability within algorithms, since the different key sizes already implied different levels of security. One objection was that not all of the ciphers could easily support round variability. Another caution was that it opened an avenue for insecure implementations in which an attacker could control the number of rounds. A third objection was that, if the AES were broken, the situation would call for analysis, not the hasty solution of, say, doubling the number of rounds. Smid pointed out that it was the cryptographic community that had asked for at least 128 bit and 256 bit key sizes, without giving guidelines on the number of rounds.

11. Future Plans and Closing

To close the conference, Smid explained how the AES process would continue and presented the following timetable. The Round 1 comment period would close April 15, 1999 and NIST would post all of the official comments on April 19, 1999. May 15, 1999 would be the deadline for the submitters to propose, if they wished, minor modifications ("tweaks") to their algorithms. Sometime in the summer of 1999, NIST would announce the finalists, beginning the Round 2 analysis. One month after the announcement would be the deadline for the submission of any updated code, which NIST would then distribute to interested parties. The deadline for papers for the third AES conference would be January 15, 2000; the conference itself would be the week of April 10, 2000. The Round 2 comment

period would close May 15, 2000. The draft AES standard, which would contain the winner(s) would be announced for public comment in late summer of 2000.

A conference feedback form was distributed to the attendees. It included an informal poll, asking the attendees which five algorithms NIST should select for Round 2, and which algorithms, if any, NIST should definitely not choose for Round 2. Smid said that the results would be posted on the AES homepage.

12. References

[1] Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard (AES), *Federal Register*, Volume 62, Number 177, September 12, 1997, pp. 48051-48058.

[2] Edward Roback and Morris Dworkin, First Advanced Encryption Standard (AES) Candidate Conference, J. Res. Natl. Inst. Stand. Technol., **104** (1), 97-105 (1998).

[3] G. Keating, Performance Analysis of AES Candidates on the 6805 CPU core, April 15, 1999, available from http://www.ozemail.com.au/%7Egeoffk/aes-6805/.

[4] P. Kocher, J. Jaffe, and B. Jun, Introduction to Differential Power Analysis and Related Attacks, Cryptography Research, Inc., available from http://www.cryptography.com/dpa/technical/.

[5] D. Whiting, J. Kelsey, B. Schneier, D. Wagner, N. Ferguson, and C. Hall, Further Observations on the Key Schedule of Twofish, Twofish Technical Report #4, March 16, 1999, available from http://www.counterpane.com/twofish.html.