Volume 104, Number 1, January–February 1999
Journal of Research of the National Institute of Standards and Technology

[J. Res. Natl. Inst. Stand. Technol. **104**, 97 (1999)]

Conference Report

FIRST ADVANCED ENCRYPTION STANDARD (AES) CANDIDATE CONFERENCE Ventura, CA August 20-22, 1998

Report prepared by

Edward Roback and
Morris Dworkin

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

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

1. Introduction

On August 20-22, 1998, 200 members of the global cryptographic research community gathered in Ventura, CA for the First Advanced Encryption Standard (AES) Candidate Conference (AES1). The conference focused on 15 cryptographic algorithms being considered for the Federal Government's Advanced Encryption Standard. Sponsored by the National Institute of Standards and Technology's (NIST) Information Technology Laboratory, AES1 provided an opportunity for the submitters of candidate algorithms to brief their proposals and answer initial questions. The purpose of the conference was to introduce participants in the analysis and evaluation process to the various candidate algorithms. This conference served as the formal kick-off of the first AES public evaluation and analysis period ("Round 1"), which runs through April 15, 1999.

2. Background and Context: the Advanced Encryption Standard Development Process

Since 1977, NIST's Data Encryption Standard (DES) [1] has been the Federal Government's standard method for encrypting sensitive information. In addition, it has gained wide acceptance in the private sector and has been implemented in a wide variety of banking applications. The algorithm specified in this standard has evolved from solely a U.S. Government algorithm into one that is used globally. However, with recent successful key exhaustive attacks, the useful lifetime of DES is now drawing to a close. Anticipating this eventuality, in 1996 NIST officials began preparing for development of a successor standard. In outlining these plans, NIST sought to construct an open process to engage the cryptographic research community and build confidence in the successor algorithm.

On January 2, 1997, NIST announced the initiation of a process to develop the AES [2], which would specify the Advanced Encryption Algorithm (AEA) and serve as an eventual successor to the venerable DES. Basic criteria that candidate algorithms would have to meet were proposed, in addition to required elements in the nomination packages to be submitted to NIST. Over thirty sets of comments were received from U.S. Government agencies, vendors, academia, and individuals. Additionally, NIST sponsored an AES workshop on April 15, 1997 to discuss the comments received and obtain additional feedback to better define the request for candidate algorithms. This input was of great assistance to NIST in preparing its formal call for algorithms and evaluation criteria.

On September 12, 1997, NIST published its formal call for algorithms. [3] Candidate algorithms had to meet three basic requirements: 1) implement symmetric (secret) key cryptography, 2) be a block cipher, and 3) support cryptovariable key sizes of 128 bits, 192 bits, and 256 bits with a block size of 128 bits. The algorithm could also support additional key and block sizes. In addition to the above requirements, submitters had to provide the following:

1. Complete written specifications of the algorithm,
2. Statements of the algorithm's estimated computational efficiency,
3. Known answer test values for the algorithm, and code to generate those values,
4. Statement of the algorithm's expected cryptographic strength,
5. Analysis of the algorithm with respect to known attacks,
6. Statement of advantages and limitations of the algorithm,
7. Reference implementation of the algorithm, specified in ANSI C,
8. Optimized implementations specified in Java^{TM} and ANSI C, and
9. Signed statements that a) identified any pertinent patents and patent applications and b) provided for the royalty-free use of that intellectual property should the candidate selected be selected for inclusion in the AES.

In its call for candidates, NIST made clear that security would be the most important criterion by which algorithms are evaluated, followed by efficiency and other characteristics. In the spirit of DES' success, NIST's goal in the AES development effort is to specify an algorithm that will have a lifetime of at least thirty years, that will be used extensively throughout the U.S. Government, and that will be also be available in the private sector, on a royalty-free basis worldwide.

Twenty-one algorithms were submitted to NIST by the June 15, 1998 deadline. After review, NIST determined that 15 of these met the minimum acceptability requirements and were accompanied by a complete submission package. These algorithms were made public by NIST on August 20, 1998 at AES1 for the first evaluation period. At the conference, submitters of the 15 candidate algorithms were invited to provide briefings on the candidates and answer any initial questions. NIST also announced its request for comments on the candidates, due April 15, 1999. These comments will help NIST narrow the field of candidates to approximately five or fewer for the second round of public evaluation. The public analysis of the candidates will be the subject of the Second AES Candidate Conference (AES2), scheduled for March 22-23, 1999. Following its study of the second round analysis, NIST intends to select one algorithm (or possibly more than one, if warranted) to be proposed for inclusion in the AES.

3. Conference Purpose, AES Development Overview, Announcement of Candidates and Review of Evaluation Criteria

Mr. Miles E. Smid, Manager of the Security Technology Group of the Computer Security Division in NIST's Information Technology Laboratory, welcomed the AES1 participants and noted that the primary purpose of the conference was to provide an opportunity for each of the submitters to formally present their candidate algorithms and design philosophy. After sketching the history of the AES development process, he noted that NIST received and reviewed 21 packages. In each case, NIST checked whether: 1) the legal documents were completed; 2) the submissions were responsive to all requirements; and 3) the given code, when run, passed the Known Answer Test." Six of the packages were incomplete; thus, 15 candidates were formally accepted into the AES development process. Mr. Smid noted NIST did not perform any cryptanalysis and, therefore, acceptance by NIST of an algorithm into the process did not signify anything regarding the strength of a candidate. A list of the six incomplete submissions was read to the audience and posted to NIST's AES website.

Mr. Smid then formally unveiled the accepted candidates, as follows:

Country of Origin	Algorithm	Submitter(s)
Australia	LOKI97	Lawrie Brown, Josef Pieprzyk, Jennifer Seberry
Belgium	RIJNDAEL	Joan Daemen, Vincent Rijmen
Canada	CAST-256	Entrust Technologies, Inc.
	DEAL	Richard Outerbridge, Lars Knudsen
Costa Rica	FROG	TecApro Internacional S.A.
France	DFC	Centre National pour la Recherche Scientifique (CNRS)
Germany	MAGENTA	Deutsche Telekom AG
Japan	E2	Nippon Telegraph and Telephone Corporation (NTT)
Korea	CRYPTON	Future Systems, Inc.
USA	HPC	Rich Schroeppel
	MARS	IBM
	RC6	RSA Laboratories
	SAFER+	Cylink Corporation
	TWOFISH	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson
UK, Israel, Norway	SERPENT	Ross Anderson, Eli Biham, Lars Knudsen

Mr. Smid then briefly reviewed the principal goals NIST has for the AES, as discussed above. The AES should be more secure and efficient than Triple DES. He noted that some of the submitters were claiming efficiency performance for their candidate to be greater than that of *single* DES.

Mr. Edward Roback, Computer Specialist in NIST's Computer Security Division, then proceeded to review the AES evaluation criteria, NIST's plans to foster discussion of the candidates, issues regarding submitting formal comments to NIST, and its plans for efficiency testing of the algorithms.

When NIST published its call for algorithms, it included a listing of the evaluation criteria by which NIST intends to make the AES selection. This was done for two reasons: 1) to aid the submitters in understanding the qualities important to NIST, and 2) to ensure that the criteria were well understood and available beforehand to avoid any possible questions of bias. There are three major categories to NIST's AES evaluation criteria: security, cost, and algorithm and implementation characteristics. Each of these has subcomponents.

Security is, of course, the paramount consideration in the AES selection process and encompasses such issues as the relative security of one candidate as compared to the others, and the extent to which the algorithm output is indistinguishable from a random permutation on the input block. Each submitter had to provide NIST with an estimate of the strength of their candidate. Therefore, any attacks demonstrating that the actual security of an algorithm is less than the claimed strength will factor into NIST's AES decision.

Cost includes licensing requirements, computational efficiency, memory requirements and flexibility. Each candidate submitter had to sign license agreements provided by NIST identifying any known intellectual property (i.e., patents or patent applications) that may be infringed by the practice of the particular candidate. If such property was identified, the owner of the intellectual property had to agree in writing to allow for its worldwide royalty-free use, should the candidate be included in the AES. (Use of the algorithms *for the purposes of AES evaluation* also had to be granted.) NIST hopes to address any other intellectual property issues that may arise during the public comment process before selecting the AES algorithm.

Computational efficiency (i.e., speed) is also a cost consideration. NIST tests the candidate algorithms on a common platform to compare performance characteristics. In the first AES evaluation round, this will focus primarily upon the 128 bit key size, while in the second round (with about five candidates), this will be expanded to include the 192 bit and 256 bit key sizes and hardware performance estimates. Memory requirements (e.g., for code, necessary memory, etc.) will also be measured. While NIST will conduct some of this analysis, it also welcomes the submission of such analysis by other parties.

Algorithm and implementation characteristics include flexibility, hardware and software suitability, and additional features offered by a candidate algorithm. For example, an algorithm may support block sizes other than the required 128 bits and key sizes other than the required 128 bits, 192 bits, and 256 bits. . Additionally, some candidates may be designed to facilitate efficient implementation on a wider variety of platforms or in diverse applications. For example, the ability to use the AES in 8 bit processor smart cards with strict memory limitations has often been cited by potential users as desirable. Simplicity of design is also a factor. If an algorithm's construction is straightforward and easier to analyze, it will likely have an edge over an unnecessarily complex design.

In order to facilitate informal discussion of the candidates and to aid NIST in following the expected on-going analysis, NIST has established electronic discussion pages for each candidate as well as other relevant AES topics (e.g., intellectual property). These are intended to aid interaction among parties evaluating particular algorithms or discussing other aspects of the AES process. It is also intended to provide a focal point for each of the 15 submitters to monitor public review of their candidates. The groups should also provide a way for evaluators to receive feedback on their ideas prior to submitting official formal public comments to NIST. Mr. Roback encouraged submitters to participate in these discussions at their discretion. NIST also welcomes suggestions for other topical discussion groups. All postings to these discussion groups will be publicly available on-line at http://www.nist.gov/aes.

Turning to NIST's solicitation of public analysis and comments of the algorithms, Mr. Roback said that NIST seeks comments on all aspects of the candidates. Comments on the algorithms as viewed against the evaluation criteria are anticipated to be the subject of a majority of the public comments. Intellectual property is another area in which comments would be useful to NIST, especially claims of intellectual property that were not known to the submitters. Analysis of the entire field of candidates would also be useful (e.g., comparison of all 15 algorithms against a particular crypt-analytic attack or efficiency testing on a common platform). Finally, NIST is seeking overall recommendations with justifying comments regarding which candidates should be selected as finalists. NIST intends to invite the submitters of particularly useful, novel or insightful comments to brief at AES2. NIST will accept formal public comments through April 15, 1999; however, comments should be received by February 1, 1999 for consideration for the AES2 program. All formal comments will be part of the official public record. E-mail comments will be accepted at "AESFirstRound@nist.gov".

In order to have *at least* one set of comparable efficiency test values for all 15 candidates, NIST will measure the efficiency of the optimized ANSI C and Java$^{\text{TM}}$ implementations on a IBM-compatible PC/Intel Pentium-pro Processor (200 MHz), with 64 MB RAM. NIST will conduct tests on other platforms with various compilers, as time and resources permit. NIST also intends to test ciphertext for randomness and to measure the timings of algorithm setup, key setup, key change, encryption, and decryption, where applicable to each algorithm. Mr. Roback emphasized that NIST is conducting these tests to ensure the existence of at least one set of efficiency measures of the entire field of candidates. Other such measurements, on different platforms, including different computer languages or using different compilers, would be welcomed by NIST.

Next, Mr. James Foti, a mathematician with NIST's Security Technology Group, explained the contents of the two CDROMs published by NIST. The first, entitled *CD-1: Documentation* contains algorithm specifications, supporting documentation, and intellectual property information. It is not subject to U.S. export controls. The second, entitled *CD-2: Algorithm Code*, contains reference and optimized algorithm code, example values, and all the information contained on CD-1. CD-2 is subject to U.S. export controls and may not be sent outside the United States or Canada without an export license. Both disks are available from NIST free of charge. Mr. Foti encouraged interested parties to see NIST's AES web site for ordering information.

4. AES Candidate Algorithm Presentations

Each submitter of a candidate algorithm accepted by NIST into the AES development process was invited to present a briefing on their submission and answer questions. The following is a summary of the presentations. The descriptions of the algorithms generally exclude the key schedules, which tend to be complicated. "Addition" refers to addition modulo the integer that corresponds to the size of the data word; moreover, "key addition [subtraction]" means modular addition [subtraction] of a round key to [from] the data word. Similarly, "key XOR" means bitwise exclusive-or of a round subkey with the data word.

CAST-256

CAST-256 is an extension of the CAST-128 cipher, using the same three round functions but generalizing the Feistel structure, so that in each round one fourth of the data block updates another fourth of the data block. There are 48 rounds, and they are constructed so that decryption is identical to encryption up to the order of the round keys. Each round function uses two types of subkeys, one to which a data block is added, subtracted, or XORed, and another that determines a rotation of the result. That in turn determines outputs of four 8×32 s-boxes which are mixed with addition, subtraction, and XOR.

The presenter, Carlisle Adams, sketched the history of the CAST family of algorithms, culminating with the endorsement of CAST-128 by the Government of Canada's Communications Security Establishment. Since CAST-256 uses the same round functions as CAST-128, it inherits 10 years of public scrutiny. He described the security contributions of the following features of the round function: the design of the bent-function-based s-boxes, the combination of a “masking” subkey and a rotation subkey, the mixing of operations from two different groups, and the mixing of the order of the group operations. He also cited the advantages of the key schedule and of the generalized Feistel structure, its symmetry, and its extensibility to other block sizes. He acknowledged minor weaknesses in simplified variants of CAST-128, such as a reduced round higher order differential attack, but said that CAST-128, and consequently CAST-256, incorporated safeguards against them.

CRYPTON

CRYPTON is a substitution-permutation network based on the design of SQUARE. There are two, alternating round functions that consist of substitution using two 8×8 s-boxes, a bit permutation followed by a byte transposition of the data array, and key XOR. There are 12 rounds, preceded by key XOR, and followed by a transformation that makes decryption identical to encryption up to the order of the round keys, which also must be suitably transformed. The two s-boxes were constructed from three 4×4 s-boxes using a three round Feistel structure.

The presenter, Chae Hoon Lim, emphasized the security of the algorithm and the efficiency and simplicity of its “fine-grained design.” The round function is fully parallelizable, so there are fast implementations in both hardware and software, almost twice as fast as DES, he claimed. He also claimed that the s-boxes were also designed to give efficient hardware implementations, as well as good linear and differential characteristics to resist those attacks and their variations. He discussed the key schedule, citing its speed, claiming that it was designed to avoid known weaknesses, but acknowledging that the designers intended to review and strengthen it. Similarly, the designers intend to construct two variants of one of the given s-boxes and incorporate them into the algorithm.

DEAL

DEAL (Digital Encryption Algorithm with Larger blocks) is a Feistel network that uses DES as its round function. For 128 bit keys or 192 bit keys there are six rounds; for 256 bit keys there are eight rounds. After the final round, the two halves of the data word are not “unswapped,” which introduces a slight asymmetry between encryption and decryption besides the order of the round keys. The key schedule expands the user key by repetition, XORs it with constant offset values, and encrypts it with DES in the Cipher Block Chaining mode under a fixed key.

The presenter, Richard Outerbridge, portrayed DEAL as a sensible evolution of the well-studied DES, surpassing the security of triple DES, and avoiding the weaknesses of DES and triple DES. The key schedule was chosen to avoid equivalent keys, related keys, and the complementation property. He emphasized that DEAL could be efficiently implemented on many platforms “almost overnight,” because DES has already been extensively deployed. He acknowledged that DEAL is at least as slow as triple DES, especially in its key setup, so DEAL is not suited for constrained environments that require dynamic rekeying. He also acknowledged a recent attack due to Lucks [4].

DFC

DFC (Decorrelated Fast Cipher) is a Feistel network with eight rounds. The round function uses multiplication and addition modulo $2^{64}+13$, reduction modulo 2^{64}, and a “confusion” permutation. This permutation uses addition modulo 2^{64} and the XOR operation with two fixed constants and another constant that is chosen from a table according to six of the data bits. Decryption is identical to encryption up to the order of the round keys.

The presenter, Serge Vaudenay, emphasized that the designers were concerned with using the recently developed technique of “decorrelation” to provide “provable security” against iterated attacks of order 2, according to a certain security model. If this could be achieved, it would imply resistance to several classes of attacks, including linear and differential ones; the designers’ strategy was to tolerate imperfect decorrelation as long as it could be quantified. He proceeded to explain their particular assumptions and the security results they achieved, forecasting, for example, that exhaustive search of an 80 bit key would require at least several decades. The documentation also cited implementations of DFC on various platforms, claiming a speed rate greater than all commercial implementations of DES.

E2

E2 (“Efficient Encryption”) is a Feistel network with 12 rounds, preceded by an initial transformation and followed by a final transformation. The initial transformation consists of key XOR, modular multiplication in 32 bit blocks with a round key, and a byte permutation; the final transformation is its inverse. The round function consists of a permutation sandwiched between two keyed substitutions, followed by a byte rotation. The permutation is a linear transformation of data bytes; each keyed substitution consists of key XOR followed by the application of an 8×8 s-box to each byte. The construction of the s-box is based on the composition of a power function in GF(2^8) and an affine function in $\mathbf{Z}/2^8$ Z. Decryption is identical to encryption up to the order of the round keys.

The presenter, Shiho Moriai, explained the rationale for the design, emphasizing the goals of security, efficiency, and flexibility. She claimed that two substitutions per round allow more speed for a given level of security than one substitution per round, and she spoke at some length about the construction and the properties of the s-box. It was constructed by mixing operations from two different groups, both to provide security against algebraic attacks and to convince the user that there are no trapdoors. She also claimed that the s-box could be efficiently implemented on many platforms, including those with 8 bit processors. She claimed that nine rounds of E2 would provide sufficient security against differential and linear attacks; the extra three rounds therefore constitute "insurance," along with the initial and final transformations, which are intended to resist new, as yet unknown, attacks.

FROG

Frog is an unconventional substitution-permutation network with eight rounds. The expanded key functions as an "interpreter" to sequentially process each byte of the data block. First, the byte is XORed with a byte of key material, and the result indexes another byte of key material. This byte in turn modifies three bytes of the data block: substituting for the original data byte, XORing with the following data byte, and XORing with a third data byte, which is also determined by key material. There is a complicated procedure for generating the large internal key from the user key. Decryption is the inverse of encryption.

The presenter, Dianelos Georgoudis, emphasized that FROG was designed under a different paradigm than conventional ciphers. Because the key determines the computational process, that process is hidden from potential attacker, and the algorithm is difficult to model mathematically. The presenter claimed, for example, that FROG resists linear and differential attacks because the substitutions are initialized with effective random values that are hidden. The other important design principle was simplicity, which, he claimed makes trapdoors and obscure structural flaws unlikely. In fact, the presenter claimed that should FROG be found to resist current methods of attack even though it was not specifically designed to do so, then one would gain confidence that it would resist future attacks, whose nature we cannot now predict. He acknowledged and discussed a recent attack due to Wagner, Ferguson, and Schneier [5].

HPC

HPC (Hasty Pudding Cipher) is a set of five subciphers, each covering a range of possible block sizes; the "medium" cipher applies to the 128 bit blocks mandated for the AES. In addition to the expansion of the user key into a lookup table, the cipher features an independent, secondary key, called the "spice," whose use and concealment are optional. The algorithm mixes these two types of key material with the data block in a complicated series of steps involving addition, subtraction, the XOR operation, fixed rotations, and data-dependent rotations. Decryption is the inverse of encryption.

The presenter, Rich Schroeppel, emphasized that HPC is an "omni-cipher"; in other words, it is flexible enough to handle variable spice size, any key size, and, especially, any block size. He said that the algorithm is "forward-looking" in that it runs best on 64-bit architectures, but, conversely, it is "smartcard hostile," and, also, "doesn't favor Pentium." He claimed that the algorithm is fast, but cited the disadvantages of the code length, the dynamic storage size, and the slow primary key setup. He acknowledged that the algorithm is inelegant and therefore hard to analyze, but nevertheless he claimed that HPC has good security.

LOKI97

LOKI97 is a based on LOKI89 and LOKI91. It varies the Feistel structure in that, both before and after the round function is applied to half of the data block, key material is added to that half. Therefore, decryption requires corresponding key subtractions as well as the usual reordering of the round keys. The round function consists of a keyed permutation, a fixed expansion function, two s-boxes, one 13×8 and the other 11×8, a fixed permutation, another expansion, this time by key material, followed by another application of the s-boxes. The s-boxes are given by cubing in GF(2^{13}) and GF(2^{11}).Decryption is similar but not identical to encryption.

The presenter, Jennifer Seberry, first mentioned some weaknesses of the predecessors to LOKI97, including attacks on reduced round versions, but claimed that the full round versions are secure. She then discussed the design goals: no simple relations, no bad keys, and resistance to linear and differential attacks. She explained the rationale behind the elements of the algorithm. The key features of the round function were the double substitution-permutation layer, the completeness property, and the hiding of the round function achieved by the extra key addition incorporated into the

Feistel structure. She cited several advantageous properties of the s-boxes. She discussed a recent attack due to Rijmen and Knudsen [6] and suggested possible changes in the algorithm for dealing with it.

MAGENTA

MAGENTA (Multifunctional Algorithm for General-purpose Encryption and Network Telecommunication Applications) is a Feistel network without "unswapping" after the final round. For 128 bit and 192 bit keys there are six rounds, and for 256 bit keys there are eight rounds. The round function acts on the bytes of the data concatenated with bytes of the round subkey. The building blocks are a fixed permutation of individual bytes, the XOR operation, and a shuffling of the bytes. The permutation is discrete exponentiation of a fixed primitive element in a given representation of GF(2^8). The round subkeys are simply disjoint 64 bit segments of the key. Because the subkeys are arranged symmetrically, decryption is almost identical to encryption, up to the swapping of the two halves of the data.

The presenter, Michael Jacobson, Jr., explained the algorithm and its algebraic properties, emphasizing the simplicity of the design. Discrete exponentiation provides the property of confusion, and he cited the transparency of the technique as an advantage over the use of s-boxes. Diffusion is provided by the shuffle structure, which is based on the fast Fourier transform. He presented analysis of the avalanche properties, other statistical properties, and the linear and differential characteristics of the round function, claiming that there are no practical linear and differential attacks. He also claimed that the algorithm is efficient in both hardware and software; he acknowledged the existence of some weak keys.

After the presentation, several attendees of the conference mounted attacks on MAGENTA based on the symmetry of the subkeys [7].

MARS

MARS is a cipher with 32 modified Feistel rounds structured as follows: key addition, eight rounds of "unkeyed forward mixing," eight rounds of "keyed forward transformation," eight rounds of "keyed backwards transformation," eight rounds of "unkeyed backwards mixing," and key subtraction. In each round, one fourth of the data word updates each of the other three fourths of the data word. The unkeyed rounds use two 8×32 s-boxes, addition, and the XOR operation. In addition to those elements, the keyed rounds use 32 bit key multiplication, data-dependent rotations, and key addition. Decryption is not identical to encryption, although it is similar in structure.

The presenter, Shai Halevi, explained the rationale for wrapping the keyed "cryptographic core" with unkeyed mixing: by providing good avalanche of the input bits, the unkeyed rounds are intended to hinder an attacker from stripping away the first and last rounds. He also claimed that this heterogeneous structure would prove resilient against new, as yet undiscovered, attacks. He cited the variety of operations, both known and new, used in the keyed rounds as another protection against future attacks. He discussed the round function of the keyed rounds in more detail, including an analysis of its linear and differential properties. He claimed that MARS offers high resistance to known attacks, better than triple DES, and runs faster than single DES in some implementations.

RC6

RC6 is a parameterized family of encryption ciphers that use a modified Feistel structure; under the parameters given for the AES submission, there are twenty rounds. The data block is partitioned into four 32 bit words. In each round, the second word updates the first word, while, in parallel, the fourth word updates the third word, after which the positions of the four words are rotated. The updating uses a quadratic transformation—requiring a 32 bit modular multiplication and addition—the XOR operation, a data-dependent rotation, and key addition. There is also key addition before the first round and after the last round. The decryption routine is derived from the encryption routine by inverting each step.

The presenter, Ron Rivest, emphasized the algorithm's simplicity, speed, and security. He explained in seven steps how the designers of RC6 adapted RC5 to meet the AES submission requirements. An important improvement was to determine the amount of the data-dependent rotations, a main source of the overall security, by the quadratic function; this method is also efficient because 32 bit multiplication is well supported on modern processors. He presented implementation results supporting his claim that RC6 is perhaps the fastest of the candidate algorithms. He cited security analysis of the algorithm, including both its resistance to linear and differential attacks and the security of the key expansion.

RIJNDAEL

Rijndael is a substitution-linear transformation network with 10, 12, or 14 rounds, depending on the key size, and with block sizes of 128 bits, 192 bits, or

256 bits, independently specified. The data block is partitioned into a 4×4, 4×6, or 4×8 array of bytes. The round function consists of three parts: a non-linear layer, a linear mixing layer, and a key XOR layer. There is also key XOR before the first round. The non-linear layer is an 8×8 s-box applied to each byte. The s-box is constructed by considering the byte as an element of GF(2^8), finding its multiplicative inverse, then applying to the corresponding vector an affine transformation over GF(2). The linear layer consists of a shifting of the rows of the array and a mixing of the columns based on maximum distance separable codes. In the last round the column mixing is omitted.

The presenter, Joan Daemen, explained the elements of the cipher: for example, he cited the diffusion properties of the linear layer, and he claimed that the s-box would be difficult to model algebraically. Although he discussed its security against a variety of attacks, he focused on the advantages of the algorithm in its implementations. There is no algorithm setup; the key schedule is fast; the code is compact; there is extensive parallelism. Thus, the algorithm runs fast on a wide range of processors, plus, he claimed, it is very flexible in hardware. He particularly mentioned its suitability for smart cards, while acknowledging that executing the inverse cipher could be twice as slow as executing the cipher there.

SAFER+

SAFER+ is a substitution-linear transformation network based on the SAFER (Secure and Fast Encryption Routines) family of ciphers. There are 8, 12, or 16 rounds, depending on the key size, plus an output transformation after the final round. The round function consists of key-controlled substitution on the sixteen bytes of the data block followed by an invertible linear transformation on the entire data block. The substitution function acts on each individual byte with a combination of key addition, key XOR, and either a fixed permutation or its inverse. The permutation corresponds to discrete exponentiation of a fixed generator in the multiplicative group of integers modulo 257. The linear transformation is generated by a combination of the Pseudo-Hadamard Transform matrix and the "Armenian Shuffle" permutation. The decryption routine is derived from the encryption routine by inverting each step.

The presenter, James Massey, explained how SAFER+ is neither a Feistel cipher nor a substitution-permutation cipher, but rather a generalization of the latter, giving the designer more freedom to seek the best properties. SAFER+ replaces the "Hadamard Shuffle" from the original SAFER family with the "Armenian Shuffle"; he claimed that this resulted in faster diffusion and better resistance to differential attacks. Some other advantages he cited were the byte orientation, the scalability of the bytes, the lack of "suspicious-looking" tables, and the mixing of additive groups. He compared C implementations of SAFER+ and DES by the same programmers to argue that the former cipher was much faster on a Pentium platform. He also claimed that SAFER+ with its eight rounds is secure against linear and differential attacks with a margin of safety, acknowledging, however, that there is no proof of complete security.

SERPENT

Serpent is a substitution-linear transformation network. It has 32 rounds, plus an initial and a final permutation to simplify an optimized implementation. The round function consists of key XOR, 32 parallel applications of the same 4×4 s-box, and a linear transformation, except in the last round, when another key XOR replaces the linear transformation. The algorithm cycles through eight different s-boxes; thus, each of them is used in four rounds. The decryption routine is the derived from the encryption routine by inverting each step.

The presenter, Eli Biham, emphasized that the designers adopted an ultra-conservative philosophy with respect to security, because the AES will need to withstand advances in both engineering and cryptanalysis for many decades. Thus they chose to base Serpent on a combination of s-boxes and linear mappings, a familiar and well-studied combination from its use in DES, and they chose to use twice as many rounds as even their conservative security analysis dictated. In addition to summarizing this analysis, the presenter described how "bitslicing" could be used to implement the algorithm efficiently, so that it would run as fast as DES.

TWOFISH

Twofish is a slightly modified Feistel network with 16 rounds. The round function acts on two 32 bit words with four key-dependent 8×8 s-boxes, followed by a fixed 4×4 maximum distance separable matrix over GF(2^8), a pseudo-Hadamard transform, and key addition. The modification to the Feistel structure is the insertion of one-bit rotations before and after the results of the round function are XORed with the other two words of the data block. This introduces a slight asymmetry between encryption and decryption besides the order of the round subkeys.

The presenter, Bruce Schneier, explained how each element of the algorithm had to meet the test of “performance driven design.” He explained how each element contributed to the security of the cipher, especially the key-dependent s-boxes. He claimed that these have an advantage over fixed s-boxes, which can be studied for weaknesses, although at the cost of longer setup times. He justified why Twofish’s process for generating s-boxes, from two fixed permutations and key material, would not yield weak s-boxes. He discussed the performance of the algorithm at length, in both hardware and software implementations. He strongly emphasized the flexibility of Twofish for many environments, citing the possibility of computing the round keys “on the fly” and of pre-computing the s-boxes to varying extents.

5. Wrap-Up and Outlook

Before adjourning, Mr. Smid expressed NIST’s appreciation to each of the submitters and acknowledged the time and effort it took to prepare an algorithm and submission package. He also thanked each for their willingness to make their algorithms available on a royalty-free basis, if selected. He expressed appreciation to the members of the cryptographic community who attended and offered their expertise for the analysis of candidates. By relying on public and private candidate algorithm submissions, soliciting public evaluation of those algorithms, and sharing its own analysis results with the public, NIST hopes to select a single algorithm for the AES that will have a high degree of public confidence from its inception. NIST is proceeding carefully but relatively rapidly, so that U.S. Government agencies will soon have a newer, stronger, and more efficient security technology available for protecting sensitive information for the next 30 years.

6. References

[1] United States Department of Commerce, National Institute of Standards and Technology Federal Information Processing Standards Publication 46-2, Data Encryption Standard (DES), December 30, 1993.

[2] Announcing Development of a Federal Information Processing Standard for Advanced Encryption Standard, Federal Register, Volume 62, Number 1, January 2, 1997, pp. 93-94.

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

[4] S. Lucks, On the Security of the 128-bit Block Cipher DEAL, http://th.informatik.uni-mannheim.de/m/lucks/papers/deal.ps.gz, August 20, 1998.

[5] D. Wagner, N. Ferguson, and B. Schneier, Cryptanalysis of Frog, http://www.counterpane.com/frog.html, August 17, 1998.

[6] V. Rijmen, L.R. Knudsen, Weaknesses in LOKI97, ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/loki97.ps.gz, June 15, 1998.

[7] E. Biham, A. Biryukov, N. Ferguson, L. Knudsen, B. Schneier, A. Shamir, Cryptanalysis of MAGENTA, http://www.counterpane.com/magenta.html, August 20, 1998.