Conference Report

THIRD ADVANCED ENCRYPTION STANDARD CANDIDATE CONFERENCE

New York, NY
April 13-14, 2000

Report prepared by Morris Dworkin

1. Introduction

On April 13-14, 2000, over two hundred members of the global cryptographic research community gathered in New York City for the Third Advanced Encryption Standard Candidate Conference (AES3). This report summarizes the conference presentations and accompanying discussions. AES3 was the third 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. At this stage of the development effort, there were five finalist candidate algorithms. The main purpose of the conference was to advise NIST in the selection of one or more of these finalists for inclusion in the standard.

The five finalist algorithms are MARS, RC6™, Rijndael, Serpent, and Twofish. MARS was submitted by the International Business Machines Corporation (U.S.). RC6 was submitted by RSA Laboratories (U.S.). Rijndael was submitted by Joan Daemen and Vincent Rijmen (Belgium). Serpent was submitted by Ross Anderson (U.K.), Eli Biham (Israel), and Lars Knudsen (Norway). Twofish was submitted by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson (U.S.).

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 (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, 192, 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 candidate algorithms. See [3] for a summary of the Second AES Candidate Conference.

2. Welcome and Overview

Edward Roback, the Chief of the Computer Security Division of NIST's Information Technology Laboratory, opened the proceedings on behalf of NIST. He welcomed the attendees to AES3 and thanked them for their participation in the process, expressing satisfaction at the turnout (230 registered participants representing at least 26 countries) and the number of papers to be presented (24). He said that he looked forward to receiving public comments, to be accepted up to May 15, 2000, and moving forward in the selection of the AES.

James Foti, a mathematician from NIST's Computer Security Division, outlined the program for the conference. There were three general conference goals: to present the Round 2 analysis of the AES candidates, to discuss relevant issues, and, especially, to provide NIST with a clearer understanding of which of the five finalist algorithms should be proposed for inclusion in the AES FIPS and which should not. The conference would address the three main criteria that NIST originally identified for evaluating the algorithms: security, efficiency, and flexibility. In addition, other issues relevant to the AES would be discussed, such as the possibility of proposing more than one algorithm for inclusion in the standard.

The conference was organized into eight sessions. On the first day, Session 1 was devoted to Field Programmable Gate Array (FPGA) evaluations; Session 2, to platform-specific evaluations; Session 3, to survey evaluations; and Session 4, to cryptographic properties and analysis. On the second day, Session 5 was a continuation of Session 4. Session 6 was devoted to a panel and audience discussion of AES issues; Session 7, to Application Specific Integrated Circuit (ASIC) evaluations and individual algorithm testing; and Session 8, to presentations from the submitters of the five finalist algorithms, followed by audience questions and discussion. In addition, Foti invited the attendees to submit proposals for short talks for a recent-results ("rump") session on the evening of the first day.

3. FPGA Evaluations

The first speaker, Adam Elbirt, spoke about the work at the Cryptography and Information Security Group at Worcester Polytechnic Institute to implement four of the five finalists (all except MARS) in FPGA hardware. Elbert first motivated the use of FPGAs for cryptographic applications: FPGA hardware was reconfigurable, so it gave flexibility to modify or replace algorithms. Moreover, for small-scale production, FPGAs were more cost effective than ASICs,

and they offered greater throughput than software. After a brief technical summary of FPGAs in general, Elbirt explained the particular technology that the group had chosen for its study: the Xilinx Virtex XCV 1000BG560-4. They deliberately chose the high-end FPGA technology of today, expecting that over the lifespan of the AES it would be a typical device.

He explained the group's design methodology. They had chosen to implement encryption (but not decryption), under 128 bit keys; they assumed that the key scheduling occurred externally and that all the subkeys were stored in internal registers; however, if necessary, the device could be reconfigured to include the key schedule or decryption functions. They used Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (HDL), i.e., VHDL, with "bottom-up" design and test methodology. He noted that the results of the study depended heavily on the synthesis and place-and-route tools, similar to the variations in software performance that occur due to compiler options or the use of C versus assembly language. The designs focused on high performance, as measured by throughput. He emphasized that if one wanted to compare the algorithms according to other measures, such as area or throughput per area, the architectures should be modified and the tools should be rerun with appropriate optimization settings.

He discussed several implementation architectures that the group had explored: iterative looping, iterative looping with partial loop unrolling, full loop unrolling, pipelining, and sub-pipelining. He then discussed the technical details for each finalist. The conclusions were that, evaluated according to throughput, Serpent exhibited by far the best performance in both feedback and non-feedback modes of operation. For non-feedback modes, the performance results for RC6, Rijndael, and Twofish were similar, but in feedback modes, Rijndael outperformed RC6 and Twofish. Each of the four algorithms in the study easily achieved Gigabit encryption rates in non-feedback modes.

Asked why MARS had not been included, Elbirt said that they had run out of time. An attendee observed that key setup time also would be important to consider; moreover, even though the implementations were optimized for throughput, area constraints did, in fact, affect the results. Following up, the chair of session, Craig Clapp, asked how specific the conclusions were to the particular chip studied. Elbirt expected that on hardware with more resources, full pipelining would be possible for RC6, Rijndael, and Twofish, and thus their performance would improve relative to Serpent. An attendee observed that Serpent might also benefit from increased area.

The second speaker of the session was Nicholas Weaver, of the Reconfigurable Architectures, Systems, and Software group of the University of California at Berkeley. The talk presented a theoretical analysis of hand-crafted designs of the finalists in hardware. Actual implementations were not used because he wanted to avoid the performance artifacts that would have been introduced by Hardware Design Language synthesis, as pointed out in the previous talk, and because specifying and laying out the datapath by hand would have taken considerable time and effort. The work was oriented toward the Xilinx Virtex FPGA family—although he claimed it applied to ASICs, as well—with a target clock cycle rate of 50 MHz.

Weaver first discussed the design factors of area, latency, bandwidth, and subkey setup, and then he summarized each finalist with respect to these criteria. He called MARS a very poor choice, because it had very poor latency, moderately heavy area requirements, and very slow key setup. He called RC6 a poor choice: although its area was reasonable, it had moderately high latency and very slow key setup. He called Rijndael a good choice, because it was reasonably compact, and it had great latency and bandwidth, although it required block RAM or similar memory, and required mostly separate pipelines for encryption and decryption. He called Serpent a fair choice. Its advantages were its very good latency and bandwidth, and it did not require block RAM. Serpent's disadvantages were high initial cost—unless at least eight rounds could be implemented, performance would suffer—key generation that was area expensive, and the requirement for separate encryption and decryption pipelines. He called Twofish a good choice, because it had good latency and bandwidth, its implementation was compact, it did not require block RAM, and it had great subkey generation.

An attendee pointed out a straightforward method for improving the efficiency of the generation of the decryption subkeys in Serpent; in response, Weaver conceded that the evaluation of subkey performance for Serpent on his summary graph should be upgraded.

The third speaker of the session was Kris Gaj, of George Mason University, who also presented a comparison of the performance of the five AES finalists in reconfigurable hardware. There were two sets of target FPGA devices: for high performance, the Virtex-XCV 1000 family; for low cost, the XC4000XL family. He presented the methodology and tools for the study: the algorithms were coded in VHDL, the code was verified, and then, as in the first study, synthesis and place-and-route simulation tools were applied. Two basic assumptions were that encryption and decryption shared as many resources as possible when this did not impose a significant speed penalty, and that all subkeys were generated on the chip and stored in memory inside of Configurable Logic Blocks, rather than embedded RAM.

After discussing the five implementation architectures that were considered for the study, Gaj presented results for both families of devices, using both feedback and non-feedback modes of operation. For feedback modes, he asserted that throughput should be the primary basis for comparison, while for non-feedback modes, in which all ciphers could achieve the same throughput, he asserted that area should be the criterion. Serpent and Rijndael's throughput in feedback modes was more than twice that of the other finalists, a result confirmed by three independent groups of researchers. For non-feedback modes, his study showed that Serpent, Twofish, and Rijndael are the most cost-efficient and require about the same area; however, there was no agreement among the three studies on methodology and architecture, so more data was needed.

4. Platform-Specific Evaluations

John Worley, of Hewlett Packard Labs, spoke on the performance of the AES finalists for actual implementations on a PA-RISC microprocessor (the PA-8500) and for simulations on a

"snapshot" of the design of the upcoming IA-64 microprocessor (McKinley chip). Both processors support multiple instructions per clock cycle. The algorithms were implemented with hand-tuned assembly code, focusing on 128 bit keys. He concluded that all of the finalists have reasonable implementations. According to his summary, Rijndael offered the best performance, followed by RC6 and Twofish, then MARS, and then Serpent. In memory usage, RC6 was ranked first (i.e., lowest), followed, in order, by Serpent, MARS, Twofish, and Rijndael. He asserted that architectures of the future would be increasingly parallel over the lifetime of the AES, so he recommended that parallelism be a factor in the performance evaluation of the finalists. For parallelism, Rijndael was judged the best, followed, in order, by Twofish, Serpent, MARS, and RC6.

The second speaker of the session was Richard Weiss, of the Compaq Computer Corporation. He presented a comparison of the AES finalists on the Alpha 21264 microprocessor, a superscalar processor that can issue four integer instructions per cycle. The study considered how the parallelism of the processor could be used to encrypt multiple, independent streams of data. For single stream timings, the five algorithms were implemented for 128 bit keys with Brian Gladman's C code, which was modified for multiple stream timings. For single stream timings, the results agreed closely with those of Granboulan: Rijndael required the fewest clock cycles, followed closely by Twofish. RC6 and then MARS were somewhat slower, followed by Serpent, which was two or three times slower than the other four algorithms. For two stream timings, the order of the algorithms was the same, but Serpent showed a significant speedup, and RC6 showed a moderate speedup.

The next speaker, Fumihiko Sano, of the Toshiba System Integration Technology Center, presented an evaluation of the AES finalists on a high-end smart card. Specifically, the finalists were implemented on the T6N55 chip, which supplements its CPU, the Z80 microprocessor, with a cryptographic coprocessor. Subkeys were generated on-the-fly for Rijndael, Serpent, and Twofish. He discussed some technical details of the implementations, and presented data for the ROM, RAM, and throughput of the finalists and also DES and Triple DES. He concluded that in ROM, Rijndael and RC6 were as small as Triple DES, and in throughput, Rijndael and Twofish exceeded Triple DES.

Thomas Wollinger, of the Cryptography and Information Security Group at Worcester Polytechnic Institute, spoke on the suitability of the AES finalists on a high-end digital signal processor (DSP), the TMS320C62x DSP. The implementations of the five finalists were coded in C and rewritten in an unrefined form of assembly language, for both single-block and multi-block modes of operation, optimized for speed, with 128 bit keys. He presented the performance data from the study. He concluded that in single-block mode, the leading performer was Rijndael, followed by RC6, and in multi-block mode, the leading performer was Twofish, also followed by RC6.

Kazumaro Aoki, of NTT Laboratories, spoke about implementations of four of the finalists (all but Serpent) on the Pentium II processor using optimized assembly language code. Speedups

over the results of other analysts ranged from 8% to 26%. The study showed that the matrix math extension (MMX) instructions available on the Pentium II could efficiently be used to speedup Rijndael, but they were only moderately useful for the other three ciphers in the study. He concluded that on the platform studied, RC6 and Rijndael were extremely fast, MARS and Twofish were very fast, and he estimated that Serpent would be fairly fast.

5. Survey Evaluations

Doug Whiting, of Hi/fn, Inc., presented an updated version of a paper that the Twofish team presented at the second AES conference, which collected and summarized software performance data on the AES finalists from various websites. Results were drawn from implementations on a variety of processor families, coded in C and assembly language, for all of the key sizes. The general trend was that, although the numbers had improved, the conclusions to draw from them were similar to the first paper. Serpent was uniformly the slowest of the finalists in software, and Rijndael and Twofish were typically among the fastest. The results for MARS and RC6 were somewhat worse than the results for Rijndael and Twofish unless the given platform supported the operations of multiplication and variable rotation. (As an aside, Whiting observed that the trend appeared to be against the inclusion of those operations in the next generation of high-speed CPUs.) For encrypting small numbers of blocks, Rijndael's performance was outstanding, because that setting magnified the effect of its fast key schedule. Rijndael's performance degraded somewhat for the higher key sizes but remained relatively fast.

Whiting also presented slides on a topic unrelated to the survey paper, namely, key agility in hardware. In some hardware applications, it might be necessary to support tens of thousands of security associations. For relatively small packets, unless the key schedule could be computed on-the-fly, the necessity to precompute and store all of the subkeys would impose a significant additional cost. The key schedules of Rijndael, Serpent, and Twofish could be computed on-the-fly, with a minimal number of gates. By contrast, he asserted that the key schedules of MARS and RC6 would have to pay performance penalties equivalent to the encryptions of 10+ and 9 blocks, respectively. An attendee pointed out that for RC6, this number could be reduced if certain "heading" states in the key schedule were precomputed.

Lawrence Bassham, of NIST's Computer Security Division, spoke about NISTs testing of the submitted C code for the finalists on a variety of combinations of processors, memory, operating systems, and compilers. He explained the methodology and presented timing results for each of the three key sizes for both encryption/decryption and key setup. He concluded with general comments on each finalist, emphasizing that these results should be weighed with results from other implementations and analysis. RC6 performed well in both key setup and encryption. For Rijndael, encryption time varied for the different key sizes, but the key setup time was clearly the best among the finalists. For MARS, key setup was average, and encryption time ranged from average to above average. Twofish had poor key setup time and average encryption times. Serpent had poor encryption times and below average key setup times.

Bassham also summarized NIST's testing of the submitted Java™ code for the finalists on the NIST reference platform, using the Java Development Kit (JDK) Version 1.3 with Just-In-Time (JIT) compilation. For encryption/decryption under 128 bit keys, Rijndael, RC6, and MARS were the fastest algorithms; for the higher key sizes, Rijndael's results dipped below those of RC6. In key setup for 128 bit keys, RC6, MARS, and Rijndael were the fastest algorithms.

The last speaker of the session was Andreas Sterbenz, of the Institute for Applied Information Processing and Communications at the University of Technology in Graz. He presented a paper analyzing the performance of the finalists for the authors' independently developed Java implementations. The code was compiled using JDK 1.1.7 with JIT, and it was run on a platform that is very similar to the NIST reference platform. RC6 had the highest throughput; MARS, Rijndael (for 128 bit keys), and Twofish had almost identical throughput; Serpent trailed significantly, although still with reasonable performance. Rijndael's performance suffered at the higher key sizes. In key setup, Rijndael was the fastest, followed, in order, by Serpent, RC6, MARS, and Twofish. He asserted that the results were in line with implementations coded in C and assembly language. Discrepancies compared to other Java studies could be attributed to inconsistent quality of the sumbitters' Java code with respect to certain optimizations.

6. Cryptographic Properties and Analysis

John Kelsey, of Counterpane Internet Security, Inc., presented several attacks on reduced-round variants of MARS. For any cipher, it was standard cryptographic practice to try to build up to attacks on the full cipher by first attacking variants with reduced numbers of rounds. In the case of MARS, he observed that it was not obvious how to define reduced-round variants because MARS's structure consisted of different kinds of rounds: 16 keyed rounds of the cryptographic core within a "wrapper" of 16 unkeyed mixing rounds and whitening. His talk focused on one of two attacks on a MARS variant consisting of 5 core rounds with the full wrapper; the attack required 2^{50} known plaintexts, 2^{247} partial encryptions, and 2^{197} bytes of memory. There were also attacks on a MARS variant consisting of 11 core rounds without any wrapper, and another variant consisting of 6 core rounds with the wrapper reduced to 6 rounds. None of the attacks were realistic to carry out. He concluded that it was difficult to evaluate the security of MARS, and the present work was a first step in developing attack methods appropriate to its structure.

Vladimir Furman, of Technion, presented two impossible truncated differentials on 8 rounds of the MARS cryptographic core. The construction was based on a 3 round truncated difference that occurred with probability 1, and the analysis used both additive differences and XOR differences. He expected that the impossible differentials could be used to attack a variant of MARS with 9 or 10 core rounds, although no such attack was described.

Tadayoshi Kohno, of Reliable Software Technology, presented attacks on reduced-round variants of Serpent. Several types of attacks were considered in the paper: meet-in-the-middle, differential, boomerang, and amplified boomerang attacks, on variants of Serpent with 7, 8, and 9 rounds. The amplified boomerang attack on the variant with 9 rounds required 2^{110} chosen

plaintexts and a work factor of 2^{252}. None of the attacks were realistic to carry out. He concluded that the results constituted only a preliminary step in the cryptanalysis of Serpent, as 9 rounds was clearly very far from the full 32 rounds.

The session on cryptographic analysis and properties continued the following morning, when Stefan Lucks, of the University of Mannheim, presented attacks on 7 round variants of Rijndael under the two larger key sizes. The attacks were based on the "Square attack," a truncated differential attack on a 6 round variant of Rijndael that was described in the Rijndael submission. Lucks extended the attack by guessing all the subkey values for an additional round and by exploiting minor weaknesses in the Rijndael key schedule. The attacks required 2^{184} and 2^{200} partial encryptions under 192 bit and 256 bit keys and the encryption of 2^{32} chosen plaintexts. Also, as Lucks had reported at the FSE2000 conference earlier in the week, his techniques could be combined with those employed by the Twofish team against Rijndael. As a result, an 8 round variant of Rijndael under 192 bit keys could be attacked, but the attack was only 16 times faster than exhaustive key search.

Henri Gilbert, of France Télécom R&D, also presented attacks on 7 round variants of Rijndael. Like the Square attack, the attacks used truncated differentials to exploit the byte oriented structure of Rijndael; however, the attacks were based on a new, efficient method for distinguishing 4 inner rounds of Rijndael from a random function. The distinguisher relied on the existence, experimentally confirmed, of expected collisions in a certain function induced by the cipher. Under the two larger key sizes, the attacks required 2^{140} operations and the encryption of 2^{32} chosen plaintexts; under 128 bit keys, a similar attack was possible, but it was only marginally faster than exhaustive key search.

Kazukuni Kobara, of the University of Tokyo, presented a method for evaluating the strength of a category of block ciphers against certains types of differential cryptanalysis. The category includes word-oriented block ciphers with small word sizes in which the linear and non-linear functions are clearly separated; Rijndael was the only AES finalist to which the analysis applied. Kobara described an efficient algorithm for estimating all truncated differential probabilities of such ciphers where randomly chosen differentials are given. He then evaluated the truncated differential probabilities of a single layer of the MixColumn operation, as well as impossible truncated differentials for multiple rounds of Rijndael. Under the assumptions of the model, impossible truncated differentials do not exist for more than 3 rounds of Rijndael.

7. Rump Session

On the first evening of the conference, several attendees gave short talks for the rump session. Andreas Dandalus, of the University of California, spoke first, presenting a comparative performance study of the finalists using FPGAs. He concluded that Rijndael and Serpent fit FPGAs the best.

Richard Schroeppel, of Sandia National Labs, presented the results of some empirical tests that he had conducted on the AES round functions, searching for correlations between various combinations of plaintext and ciphertext bits. As expected, for each algorithm, he did not detect any correlations after just a couple of iterations of the round function.

Bruce Schneier, of Counterpane Internet Security, Inc., presented some of the history of the cryptanalysis for three block ciphers (DES, IDEA, and RC5) in order to illustrate how attacks generally improve over time. He suggested that attacks on reduced-round variants, even though they are wildly impractical, give some indication of whether a practical attack is ever likely to be developed.

Steve Bellovin, of AT&T Labs, contended that IPSec (Internet Protocol Security) was a major application to consider in evaluating the key agility of the AES finalists, as big servers with many clients could not afford a lot of cryptographic overhead for short sessions.

Lars Knudsen, of the University of Bergen, retracted the truncated differential attack on Twofish that he had presented at the rump session for the FSE2000 conference earlier in the week. He discussed how the attack might be rehabilitated.

Neils Ferguson, of Counterpane Internet Security, Inc., discussed a class of keys, which he called "semi-equivalent," for certain reduced-round variants of MARS, under which these MARS variants exhibited a property that was analogous to the complementation property of the DES.

Ferguson gave a second talk in which he claimed that the results of experiments on a simplified variant of Twofish confirmed that Knudsen's retracted attack indeed would not work.

Antoine Joux, of the SCSSI (Service Central de la Sécurité des Systèmes d'Information), spoke about the role of the data-dependent rotations in the pseudo-randomness of idealized variants of RC6.

Matt Robshaw, of RSA Laboratories, disputed some of the estimates and conjectures in the MARS submission concerning MARS's resistance to linear cryptanalysis.

Brian Gladman discussed his implementations of the AES finalists on an ARM processor coded in both C and assembly language. He observed that Serpent mapped extremely well in C code but performed relatively worse in register-poor environments.

Craig Clapp, of PictureTel Corporation, discussed the performance of the AES finalists on the TriMedia VLIW media-processor, which was capable of considerable instruction-level parallelism. He concluded that on advanced CPUs, the relative performance of the algorithms may differ wildly, depending on the mode of operation: Rijndael's inherent parallelism allowed the best performance for non-feedback modes, while RC6 had the best performance for non-feedback (interleaved) modes.

8. AES Issues

Miles Smid, of CygnaCom Solutions, chaired a session devoted to the discussion of issues related to the selection of the AES. In particular, one important unresolved issue was the possibility of "multiple winners," i.e., whether NIST should propose more than one of the finalist algorithms for inclusion in the AES FIPS. Both of the papers that would be presented in the session advocated some form of multiple winners. In the interest of balance, Smid pointed out some arguments for a single winner. For example, a single winner would seem appropriate if one finalist algorithm were deemed clearly superior to the others in security, and at least equal to the others in the other criteria. Also, a single winner offered the advantages for interoperability and ease of implementation. After the two presentations, the attendees of the conference would have the opportunity to express their views in an open discussion.

Don Johnson, of Certicom Corporation, gave the first presentation. He began by pointing out that, in a later session, the submitters would all make a case for their own algorithms for the AES. (As an aside, he wondered which algorithms besides their own they favored, and why.) Therefore, he implied, the AES might be most useful as a "crypto toolbox" that contained multiple algorithms; the needs of particular applications could then determine the most appropriate algorithm. Another factor that favored multiple algorithms was his criterion of "future resilience," the ability of the eventual AES to respond to an uncertain future of cryptographic developments. Among other examples of this uncertain future, he cited various conceivable breakthroughs in quantum computing.

Johnson summarized the main arguments in favor of a single algorithm as simplicity, interoperability, and cost-effectiveness in implementation and testing. He then offered several additional arguments for multiple algorithms. The extension of cryptographic knowledge would be promoted. Some applications might require a "super AES" in which multiple algorithms were combined. Potential intellectual property disputes might be mitigated. "Target diffusion" might reduce the resources that an adversary would be willing to commit to attacking any one algorithm. NIST should avoid artificial tiebreakers in the selection process, especially since the selection was a multi-dimensional problem with imperfect information. Hardware that was deployed with a single algorithm might be difficult to change in the event of a problem, and, in general, developing a substitute algorithm might take years. The deployment of a single algorithm could result in an over-optimized infrastructure, as he claimed occurred with the DES, with the result that, in some cases, products lacked flexibility for any other algorithms. Rather than NIST choosing a single finalist for almost all settings, the marketplace might be better equipped to choose from among, say, two or three of the finalists.

Ian Harvey, of nCipher Corporation, was the other speaker of the session. He observed that there did not yet exist any significant security results to distinguish the candidates; yet the finalists offered different performance advantages/tradeoffs, so the choice of a single algorithm at this time would appear to be arbitrary. His paper discussed three different options for the selection of

two or more algorithms, under which it would be possible to maintain interoperability. In Option A, all implementations would be required to support all of the selected algorithms. In Option B, all implementations would be required to support one primary algorithm; any of the other selected algorithms could be implemented optionally as backups. In Option C, all implementations would be required to support a majority of the selected algorithms, ensuring that any two implementations would be interoperable. He evaluated each of the options against the following five criteria: security (theoretical and practical), performance (speed and resource requirements), cost of implementations, architectural implications, and legal/intellectual property issues. He concluded that all of the options increased security, increased speed performance (but not size), and created architectural issues; Option B was the best option because costs were not necessarily increased. He suggested that the primary algorithm in Option B should be determined mostly by the criterion of security; small size could be considered an advantage, but speed need not be considered an advantage, because speed would be possible to obtain from the backup algorithm(s).

In the ensuing discussion, almost all of the attendees who spoke favored the selection of a single algorithm for the AES. The idea of an optional backup algorithm of some form did not draw explicit opposition, other than the assertion that the fielding of any additional algorithms in products would increase the exposure of the AES to intellectual property disputes.

9. ASIC Evaluation and Individual Algorithm Testing

Tetsuya Ichikawa, of Mitsubishi Electric Engineering Co., presented the results of hardware simulations of implementations of the AES finalists, along with DES and Triple-DES for comparison. The design goal was to evaluate the fastest possible encryption speed, in feedback modes, under common, fair implementation conditions. The key size was 128 bits, and registers were provided for the storage of all subkey bits. The architectures were fully loop-unrolled but not pipelined, i.e., there were no storage registers between rounds. The design language was Verilog-HDL, using Mitsubishi Electric's 0.35 micron Complementary Metal Oxide Semiconductor (CMOS) ASIC design library. The simulator was the Verilog-XL, and the logic synthesis tool was the Synopsys Design Compiler, version 1998.08. Arithmetic operations were taken from the Synopsys Design Ware Basic Library, so that the multiplication operations in MARS and RC6 were not highly optimized. Similarly, the algorithms' lookup tables were not optimized with special techniques; for example, the bitslice implementation of Serpent was not used.

The algorithms were evaluated according to their maximum worst-case (maximum) throughput. In Ichikawa's opinion, the average case, which was often fifty percent faster, was too optimistic to apply to actual ASIC hardware. Rijndael had the greatest throughput, almost 2 Gb/s, followed by Serpent and DES, each at about 1 GB/s. The throughput for both Twofish and Triple DES was about 400 Mb/s, and, for MARS and RC6, about 200 Mb/s. He presented the fraction of the critical path that was required for the various constituent operations: S-box lookups, linear transformations, additions, multiplications, rotations, etc. He also presented results on key setup:

Twofish, Rijndael, and Serpent were the fastest, requiring less time than one encryption in each case; by contrast, the key setups of MARS and RC6 took more than twice as long as single encryptions.

The second speaker of the session was Bryan Weeks, of the National Security Agency (NSA), which also conducted hardware simulations of the AES finalists, with the goal of providing NIST with an unbiased hardware evaluation of the algorithms. In each case, two different architectures, iterated and pipelined, were studied. The iterated architecture required less chip area and was targeted to current virtual private network (VPN) business and web applications. The pipelined architecture yielded higher throughput and was targeted to future high-speed network encryptors or VPNs. In both cases, decryption was included along with encryption. Most of the data in his presentation was generated from implementations that supported all three key sizes together; data on separate implementations for each key size would be included in the final study. The implementations were modeled in VHDL using a 0.50 micron CMOS ASIC design library. The simulator was the Synopsis VHDL System Simulator, and the logic synthesis tool was the Synopsys Design Compiler, version 1999.10.

Weeks presented data from the simulations for a variety of performance measures of the algorithms. For both architectures, Rijndael and Serpent achieved significantly higher throughputs than the other three algorithms. In the iterative architecture, Rijndael's throughput was higher than Serpent's, at the cost of high area; Twofish, RC6, and Serpent required the least area, not counting the key schedules. In the pipelined architecture, Serpent's throughput was highest at a moderate cost in area; RC6 and Twofish required the least area. For both architectures, the key setup performance of MARS and RC6 was significantly worse than the other three algorithms in both area and time. Weeks then presented some observations on the functions within each individual algorithm, in particular, their contributions to the time in the critical path, and the fraction of the total area of that the functions required. He concluded that the performance of the algorithms varied across the parameters, so preferences among the algorithms depended on the weight that NIST and the cryptographic community assigned to the parameters.

The third speaker of the session was Akashi Satoh, of the International Business Machines Corporation (IBM). He presented an optimized design for MARS in hardware, written in VHDL using IBM's CMOS library, and synthesized using 0.18 micron copper CMOS standard cell technology. The design included high-speed customized adders and multipliers. The study showed that MARS could perform well in ASICs, achieving, in the best-case, throughput of 1.28 Gbit/sec for non-feedback modes, and requiring 13,800 gates, plus 2.25 Kbytes of memory. Although some of the other finalists could perform even better, such gains in speed or size were minor compared to the importance of the long-term security, especially in light of continually improving semiconductor technology. (He admitted that MARS was not well suited to FPGA implementations, but he claimed that, for security reasons, the AES should not be implemented in reconfigurable hardware.)

The last speaker of the session, Dag Arne Osvik, of the University of Bergen, presented an optimization of the calculation of Serpent's S-boxes on the x86 family of processors. His optimization was a refinement of the submitters' "bitslice implementation," in which the S-boxes were calculated by sequences of Boolean operations rather than lookup tables. Osvik explained his method for finding sequences that were tailored to the capabilities of the x86 processors. For example, his instruction sequences required the use of only five of the eight available registers; by contrast, register pressure was typically not a problem in reduced instruction set computer (RISC) processors. As a result, he significantly increased Serpent's throughput on the Pentium and Pentium Pro processors, including speedups of the key schedule. He also noted the implications of his work for other settings, such as the IA64 processor and ASICs; in general, he concluded that the best implementations of the S-boxes would be tailored to the underlying architecture.

10. Algorithm Submitter Presentations

Smid moderated the final session devoted to presentations from the algorithm submitters, followed by questions from the conference attendees.

Vincent Rijmen, of Katholieke Universiteit Leuven, spoke on behalf of the Rijndael submitters. He began with two observations on security: first, that none of the finalists had been broken, and, second, that Rijndael's operations lent themselves to security against implementation attacks such as differential power analysis (DPA). He presented a table of the efficiency of the other four finalists, measured in "equivalent Rijndael rounds," on the Pentium and 6805 processors, for the encryption of single blocks, four blocks, and many blocks. Except the entry for the encryption of many blocks using RC6 on the Pentium processor (9 rounds), every entry in the table was larger, and usually far larger, than Rijndael's 10 rounds. He discussed several aspects of the design philosophy: simplicity, symmetry, mutual independence of design elements, parallelism, and extendibility of the block size, key size, and number of rounds. He asserted that Rijndael, with its light key schedule and its lack of arithmetical operations and data-dependent rotations, could be expected to perform well on the unknown computing platforms of the future. He concluded with the observation that in all of the studies, on a variety of platforms, Rijndael's performance was usually among the best of the finalists and never problematic: its versatility made it a good choice for the AES.

Ross Anderson, of Cambridge University, spoke on behalf of the Serpent submitters. He interpreted NIST's evaluation criteria as a call for "the most secure algorithm which was also acceptably fast (i.e., faster than triple DES) on a wide range of platforms"; Serpent had been engineered exactly to meet this requirement. He discussed the protection requirements of the AES, claiming that it would need to remain secure for more than a hundred years. Serpent was designed for such longevity: its structure was simple and easy to analyze, its primitive operations were well-understood, and it had many more rounds than were needed today. Serpent's simplicity also facilitated its proper implementation and its proper use within systems (Anderson also recommended 256 bit keys as the default setting for the AES). He said that for any of the other

finalists, there was a risk of a "certificational" attack in the lifetime of the AES; he argued that such an attack would affect public confidence, with tangible economic consequences, as occurred with the differential attacks on the DES. He asserted that Serpent's security was not achieved at an unacceptably high cost in speed: in fact, Serpent was best in hardware, second on smartcards, and, for the most likely critical applications of the future, second on IA-64 and PA/RISC processors. Even for the encryption of large files using the Pentium processor, Serpent more than met NIST's call for an algorithm that was faster than triple DES. He concluded that Serpent should be chosen for the AES because it was the most secure of the finalists.

Shai Halevi, of IBM, spoke on behalf of the MARS submitters. He focused on security, the primary design goal. He cited the comprehensive security analysis of MARS, from the unbalanced Feistel structure to the choice of operations, and he asserted that the MARS team's security estimates had been supported by the results of other analysts. He disputed the idea that MARS was too complex, noting that MARS required the second fewest lines of C code in Gladman's implementations of the finalists, and presenting pseudo-code for all of the rounds of MARS on a single page. In connection with security margin, he asserted that all of the finalists were secure absent a breakthrough in cryptanalysis, which would render current bounds useless. MARS was unique among the finalists in its use of many "fail-stop mechanisms" to protect against such future breakthroughs. He concluded with a few general observations on each finalist. He gave plusses for performance to Rijndael (only for 128 bit keys), RC6, Twofish, but he gave each a minus for some aspect of their security. Specifically, he claimed that RC6's reliance on data-dependent rotations constituted a "single point of failure," that Rijndael's algebraic structure might lead to an attack, and that Twofish's key-dependent S-boxes were difficult to analyze. He gave Serpent a plus for security margin but a minus for performance. He concluded that MARS, with its hybrid structure, careful analysis, time-tested operations, and balance of security with performance, was the best choice for the AES.

Ron Rivest, of the Massachusetts Institute of Technology, spoke on behalf of the submitters of RC6. First, he presented RC6 in twelve lines of code (excluding the key schedule). He emphasized RC6's simplicity: it allowed for a rapid understanding of RC6's security and made analysis straightforward, in contrast with MARS and Twofish; it also made RC6 easy to implement, allowing compilers to produce high quality code without complicated optimizations. He discussed how the extensive analysis of RC6 benefited from the analysis of simplified variants and of its predecessor, RC5. He touted the security of RC6's key schedule, which had been studied for over six years. In general, he asserted that RC6 provided a solid, well-tuned margin for overall security, observing that independent analyses supported RC6's original security estimates. He then discussed RC6's performance: for 32 bit processors, excellent; for smart cards, well suited to some, and fit on cheaper ones; for 64 bit processors, generally good, with dramatic gains from multi-block processing. He identified as major trends the increasing use of Java and DSP chips; RC6 performed excellently in both. He noted the flexibility provided by RC6's parameterization of key size, rounds, and block size. In evaluating the five finalists, he recommended as an exercise the assignment of quantitative scores, and he presented one possibility for weighting the performance criteria. He concluded that RC6, with its good

performance on most important platforms, its simplicity to code, its flexibility, and its status as the most studied and best understood finalist, was the secure and elegant AES choice.

Bruce Schneier, of Counterpane Internet Security, Inc., spoke on behalf of the Twofish submitters. He explained how Twofish had been designed from the beginning with efficiency in mind. He observed that in the various performance surveys, Twofish's results are never bad, which was important because the future importance of the various platforms was unknown. He touted Twofish's unique ability to trade off encryption speed for key setup, tuning Twofish to the application, depending on the size of the data to be encrypted. He also liked the property (shared by Rijndael) that Twofish specified more work for the larger key sizes. He preferred to call Twofish modular rather than complex because, at a high level, there were only a few elements, which could be abstracted to aid analysis, while the structure within these elements allowed for great flexibility in implementation. He discussed why the key separation property of Murphy did not pose a serious threat. He cautioned against relying on some of the individual performance charts, but he suggested that the general performance of the finalists was clear. In software, Rijndael and Twofish were best, with the edge to Rijndael; MARS and RC6 were good where their rotation and multiplication operations were supported; Serpent performed much less well. In hardware, Rijndael and Serpent were the best; Twofish was good to average, and better where key agility was important; RC6 and MARS were fair to poor. He recommended that NIST choose either Rijndael extended to 18 rounds, Serpent, or Twofish for the AES; he favored Twofish for its efficiency across the board, its unique flexibility, and its speed-security tradeoff.

11. Audience Questions and Answers and Remaining Issues for the AES Development Effort

The comments that were offered in the question and answer session are summarized below according to the topic, not necessarily in chronological order.

The panelists were asked which algorithm, other than their own, they would choose for the AES. Rijmen liked RC6; the other four panelists said Rijndael if it was extended to 18 or more rounds.

Two questions were addressed to particular panelists. Asked to comment on the suggestion to increase the number of rounds of Rijndael, Rijmen did not see any need for an increase, but he observed that performance would remain decent. Asked to comment on the key agility and hardware performance of RC6, Rivest did not think that RC6 performed badly. However, he observed that RC6's performance improved dramatically relative to the other finalists in pipelined versions and that its multiplication operation could be optimized, so he cautioned against taking the current surveys as the final word.

The panelists were solicited for advice for NIST in carrying out the process. Anderson urged NIST to stick to its original specification and to consider the likely platforms for the 21st century. Rivest repeated his preference for a primary algorithm and a backup algorithm, and he asked NIST to encourage continuing analysis of whatever algorithm was chosen. Schneier advised

NIST to consider the public comments carefully, and to choose a single algorithm (although a backup was okay). Halevi advised NIST to keep a sense of perspective and realize that any of the finalists were likely to be acceptable. An attendee added that NIST should announce its choice before there could be political interference.

The possibility of multiple algorithms was discussed. Halevi strongly favored one algorithm, even if it was not MARS; Rijmen also favored one algorithm. Rivest favored a primary and a backup algorithm that would only be used if the primary algorithm was broken. He was particularly concerned about intellectual property attacks, so he disagreed with the suggestion to use a variant of the primary algorithm with increased rounds as a backup. An attendee recommended that compliance with the standard should be possible without actually fielding the backup in products, so that if the backup algorithm were challenged, the standard would not have to be withdrawn. Roback observed that the current (non-binding) precedent in standards like the Digital Signature Standard (DSS) was to require the implementation of only one of the specified options. Schneier agreed that a backup was a good idea: although the intellectual property threat would decay over time, cryptographic attacks tended to have the opposite profile. The three Serpent submitters had different opinions on multiple algorithms, but Anderson could support Serpent as the primary algorithm, with a backup that was chosen for its performance on the popular platforms of the day. An attendee observed that, if the primary algorithm were broken, it would be unwise to plan to switch immediately to a backup, because it would require analysis to determine if the attack also threatened the backup. Another attendee suggested that a backup could be named but not fielded unless it were needed; nevertheless, he urged that mechanisms be provided for switching algorithms.

The panelists were asked how they defined an appropriate security level. Halevi observed that in any system, the algorithm was not likely to be the security bottleneck, and from this point of view all of the finalists were adequate. A different point of view was that even a theoretical attack would be psychologically devastating, so 256 bit security should be the target. Schneier observed that there was no good theory of security, so, after thorough analysis, setting the security level was an ad hoc process, a "best guess." Halevi added that designers should provide protection in case their best guesses were wrong. Anderson responded to a suggestion to transfer some of Serpent's rounds to Rijndael by reiterating his support for 32 round Serpent with 256 bit keys. He observed that attacks based on too few rounds were orthogonal to attacks based on too few key bits; if lower numbers of either rounds or key bits were permitted, then NIST ought to issue warnings about their use. Schneier said that even when the cryptography of a fielded system was questionable, other aspects of its security were almost always worse.

An attendee asked if there was enough cryptanalysis to create confidence in any choice. Schneier and Rijmen answered that it was impossible to be sure that an algorithm was secure. Rivest added that the shortness of the process favored the simpler algorithms, especially Serpent and RC6, but repeated his suggestion that NIST foster more analysis after the AES is established. Anderson agreed, and mentioned that the Serpent designers chose to employ well-understood primitive operations, against which progress in cryptanalysis was relatively less likely.

The attendees were provided updates on several intellectual property questions. MARS, Rijndael, Serpent, and Twofish would be available royalty free even if they were not chosen as the AES; Rivest was not sure about RC6. In fact, the submitters of Rijndael, Serpent, and Twofish did not hold any patents to enforce. Halevi and Rivest could not speak for IBM and RSA Labs on the use of variable rotations in MARS and RC6, nor on the status of royalties for variants with different numbers of rounds. Roback reported that NIST was in the process of following up on a letter received recently from Hitachi, Inc., which indicated that Hitachi had patents that might be relevant to the AES selection.

Attendees also discussed several implementation issues. The suggestion that NIST favor designs that were straightforward to implement correctly and optimally was countered by the claim that expert code for the AES would be available for the major platforms; similarly, experts would design AES hardware. It was suggested that NIST specify carefully the endianness and the ordering of bit patterns in the AES FIPS. NIST was also advised to develop a standard for the formatting of data prior to encryption, and NIST was asked whether new modes of operation for the AES were planned. It was pointed out that new modes of operation required extensive consideration, and NIST was asked whether a conference could be held to promote their study. Roback indicated that NIST was open to holding a workshop on modes of operation and welcomed comments on these issues.

12. Future Plans and Closing

To close the conference, Roback explained how the AES process would continue and presented the following tentative timetable. The Round 2 comment period would close May 15, 2000 and NIST would post all of the official comments on May 16, 2000. NIST would announce the selected algorithm(s), and release its summary report on the selection, in the summer or fall of 2000. Soon thereafter, a draft AES FIPS specifying the winner(s) would be announced for public comment, and perhaps a draft modes of operation FIPS. Those FIPS would be slated for approval by the spring or summer of 2001, along with a validation guideline for the modes of operation FIPS. Roback concluded by thanking all of the people involved with the conference, particularly Smid, formerly of NIST, who played a leading role in the AES development effort.

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] "Conference Report: First Advanced Encryption Standard (AES) Candidate Conference," *Journal of Research of the National Institute of Standards and Technology*, Volume 104, Number 1, January-February 1998, pp. 97-105.

[3] "Conference Report: First Advanced Encryption Standard (AES) Candidate Conference," *Journal of Research of the National Institute of Standards and Technology*, Volume 104, Number 1, January-February 1998, pp. 97-105.

NIST's AES Homepage

http://www.nist.gov/aes
