89

A Look at the Advanced Encryption Standard (AES)

Ben Rothke

In the early 1970s, the Data Encryption Standard (DES) became a Federal Information Processing Standard[1,2] (FIPS). This happened with little fanfare and even less public notice. In fact, in the late 1960s and early 1970s, the notion of the general public having an influence on U.S. cryptographic policy was utterly absurd. It should be noted that in the days before personal computers were ubiquitous, the force of a FIPS was immense, given the purchasing power of the U.S. government. Nowadays, the power of a FIPS has a much lesser effect on the profitability of computer companies given the strength of the consumer market.

Jump to the late 1990s and the situation is poles apart. The proposed successor to DES, the AES, was publicized not only in the *Federal Register* and academic journals, but also in consumer computing magazines and the mainstream media.[3]

The entire AES selection process was, in essence, a global town hall event. This was evident from submissions from cryptographers from around the world. The AES process was completely open to public scrutiny and comment. This is important because, when it comes to the design of effective encryption algorithms, history has shown time and time again that secure encryption algorithms cannot

[1]FIPS 46-3, see http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf. Reaffirmed for the final time on October 25, 1999.

[2]Under the Information Technology Management Reform Act (Public Law 104–106), the Secretary of Commerce approves standards and guidelines that are developed by the National Institute of Standards and Technology (NIST) for federal computer systems. These standards and guidelines are issued by NIST as Federal Information Processing Standards (FIPS) for use government wide. NIST develops FIPS when there are compelling federal government requirements, such as for security and interoperability, and there are no acceptable industry standards or solutions.

[3]While IBM and the U.S. government essentially designed DES between them in what was billed as a public process, it attracted very little public interest at the time.

AU7495—Chapter89—29/1/2007—15:00—VELU—14749—XML MODEL CRC12a – pp. 1151–1158.

be designed, tested, and verified in a vacuum. In fact, if a software vendor decides to use a proprietary encryption algorithm, that immediately makes the security and efficacy of the algorithm suspect[4] Prudent consumers of cryptography will *never* use a proprietary algorithm.

This notion is based on what is known as Kerckhoff's assumption[5]. This assumption states the security of a cryptosystem should rest entirely in the secrecy of the key and not in the secrecy of the algorithm. History has shown, and unfortunately, that some software vendors still choose to ignore the fact that completely open-source encryption algorithms are the only way to design a truly world-class encryption algorithm.

89.1 The AES Process

In January 1997, the National Institute of Standards and Technology (NIST, a branch within the Commerce Department) commenced the AES process[6] A replacement for DES was needed due to the ever-growing frailty of DES. Not that any significant architectural breaches were found in DES; rather, Moore's law had caught up with it. By 1998, it was possible to build a DES-cracking device for a reasonable sum of money.

The significance of the availability of a DES-cracking device to an adversary cannot be understated because DES is the world's most widely used, general-purpose cryptosystem. For the details of this cracking of DES,[7] see *Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design* by the Electronic Frontier Foundation (1998, O'Reilly & Assoc.).

DES was reengineered and put back into working order via the use of Triple-DES. Triple-DES takes the input data and encrypts it three times. Triple-DES (an official standard in use as ANSI X9.52-1998[8]) is resilient against brute-force attacks, and from a security perspective, it is adequate. So why not simply use Triple-DES as the new AES? This is not feasible because DES was designed to be implemented in hardware and is therefore not efficient in software implementations. Triple-DES is three times slower than DES; and although DES is fast enough, Triple-DES is far too slow. One of the criteria for AES is that it must be efficient when implemented in software, and the underlying architecture of Triple-DES makes it unsuitable as an AES candidate.

The AES specification called for a symmetric algorithm (same key for encryption and decryption) using block encryption of 128 bits in size, with supporting key sizes of 128, 192, and 256 bits. The algorithm was required to be royalty-free for use worldwide and offer security of a sufficient level to

[4]See B. Schneier, Security in the Real World: How to Evaluate Security Technology, *Computer Security Journal*, 15(4), 1999; and B. Rothke, Free Lunch, *Information Security Magazine*, Feb. 1999, www.infosecuritymag.com.

[5]There are actually six assumptions. Dutch cryptographer Auguste Kerckhoff wrote *La Cryptographie Militare* (Military Cryptography) in 1883. His work set forth six highly desirable elements for encryption systems:

a. A cipher should be unbreakable. If it cannot be theoretically proven to be unbreakable, it should at least be unbreakable in practice.

b. If one's adversary knows the method of encipherment, this should not prevent one from continuing to use the cipher.

c. It should be possible to memorize the key without having to write it down, and it should be easy to change to a different key.

d. Messages, after being enciphered, should be in a form that can be sent by telegraph.

e. If a cipher machine, code book, or the like is involved, any such items required should be portable and usable by one person without assistance.

f. Enciphering or deciphering messages in the system should not cause mental strain, and should not require following a long and complicated procedure.

[6]http://csrc.nist.gov/encryption/aes/pre-round1/aes_9701.txt.

[7]Details are also available at www.eff.org/descracker.html.

[8]The X9.52 standard defines triple-DES encryption with keys k_1, k_2 and k_3; k_3 as: $C = Ek_3\ (D_{k2}\ (E_{k1}\ (M)))$ where E_k and D_k denote DES encryption and DES decryption, respectively, with the key k.

protect data for 30 years. Additionally, it must be easy to implement in hardware as well as software, and in restricted environments (i.e., smart cards, DSP, cell phones, FPGA, custom ASIC, satellites, etc.).

AES will be used for securing sensitive but unclassified material by U.S. government agencies.[9] As a likely outcome, all indications make it likely that it will, in due course, become the de facto encryption standard for commercial transactions in the private sector as well.

In August 1998, NIST selected 15 preliminary AES candidates at the first AES Candidate Conference in California. At that point, the 15 AES candidates were given much stronger scrutiny and analysis within the global cryptography community. Also involved with the process was the National Security Agency (NSA).

This is not the place to detail the input of the NSA into the AES selection process, but it is obvious that NIST learned its lesson from the development of DES. An initial complaint against DES was that IBM kept its design principles secret at the request of the U.S. government. This, in turn, led to speculation that there was some sort of trapdoor within DES that would provide the U.S. intelligence community with complete access to all encrypted data. Nonetheless, when the DES design principles were finally made public in 1992,[10] such speculation was refuted.

89.2 The AES Candidates

The 15 AES candidates chosen at the first AES conference are listed in Exhibit 89.1.

A second AES Candidate Conference was held in Rome in March 1999 to present analyses of the first-round candidate algorithms. After this period of public scrutiny, in August 1999, NIST selected five algorithms for more extensive analysis (see Exhibit 89.2).

In October 2000, after more than 18 months of testing and analysis, NIST announced that the Rijndael algorithm had been selected as the AES candidate. It is interesting to note that only days after NIST's announcement selecting Rijndael, advertisements were already springing up stating support for the new standard.

In February 2001, NIST made available a Draft AES FIPS[11] for public review and comment, which concluded on May 29, 2001.

This was followed by a 90-day comment period from June through August 2001. In August 2002, NIST announced the approval of Federal Information Processing Standards (FIPS) 180-2, Secure Hash Standard, which contains the specifications for the Secure Hash Algorithm (SHA-1, SHA-256, SHA-384, and SHA-512).

89.2.1 DES Is Dead

It is clear that not only is 56-bit DES ineffective, it is dead. From 1998 on, it is hoped that no organization has implemented 56-bit DES in any type of high-security or mission-critical system. If such is the case, it should be immediately retrofitted with Triple-DES or another secure public algorithm.

Although DES was accepted as an ANSI standard in 1981 (ANSI X3.92) and later incorporated into several American Banking Association Financial Services (X9) standards, it has since been replaced by Triple-DES.

Replacing a cryptographic algorithm is a relatively straightforward endeavor because encryption algorithms are, in general, completely interchangeable. Most hardware implementations allow plug-ins and replacements of different algorithms. The greatest difficulty is in the logistics of replacing the software for companies with tens or hundreds of thousands of disparate devices. Also, for those organizations that have remote sites, satellites, etc., this point is ever more germane.

[9] It should be noted that AES (like DES) will only be used to protect sensitive but unclassified data. Classified data is protected by separate, confidential algorithms.

[10] Dan Coppersmith, The Data Encryption Standard and Its Strength Against Attacks, IBM Report RC18613.

[11] http://csrc.nist.gov/encryption/aes/draftfips/fr-AES-200102.html.

EXHIBIT 89.1 AES Candidates Chosen at the First AES Conference

Algorithm	Submitted By	Overview[a]
CAST-256	Entrust Technologies, Canada	A 48-round unbalanced Feistel cipher using the same round functions as CAST-128, which use +—XOR rotates and 4 fixed 6-bit S-boxes; with a key schedule.
Crypton	Future Systems, Inc., Korea	A 12-round iterative cipher with a round function using & | XOR rotates and 2 fixed 8-bit S-boxes; with various key lengths supported, derived from the previous SQUARE cipher.
DEAL	Richard Outerbridge (UK) and Lars Knudsen (Norway)	A rather different proposal, a 6- to 8-round Feistel cipher which uses the existing DES as the round function. Thus a lot of existing analysis can be leveraged, but at a cost in speed.
DFC	Centre National pour la Recherche Scientifique, France	An 8-round Feistel cipher design based on a decorrelation technique and using +x and a permutation in the round function; with a 4-round key schedule.
E2	Nippon Telegraph and Telephone Corporation, Japan	A 12-round Feistel cipher, using a nonlinear function comprised of substitution using a single fixed 8-bit S-box, a permutation, XOR mixing operations, and a byte rotation.
FROG	TecApro International, South Africa	An 8-round cipher, with each round performing four basic operations (with XOR, substitution using a single fixed 8-bit S-box, and table value replacement) on each byte of its input.
HPC	Rich Schroeppel, United States	An 8-round Feistel cipher, which modifies 8 internal 64-bit variables as well as the data using +—x & | XOR rotates and a lookup table.
LOKI97	Lawrie Brown, Josef Pieprzyk, and Jennifer Seberry, Australia	A 16-round Feistel cipher using a complex round function f with two S–P layers with fixed 11-bit and 13-bit S-boxes, a permutation, and + XOR combinations; and with a 256-bit key schedule using 48 rounds of an unbalanced Feistel network using the same complex round function f.
Magenta	Deutsche Telekom, Germany	A 6- to 8-round Feistel cipher, with a round function that uses a large number of substitutions using a single fixed S-box (based on exponentiation on $GF(2^8)$), that is combined together with key bits using XOR.
MARS	IBM, United States	An 8+16+8-round unbalanced Feistel cipher with four distinct phases: key addition and 8 rounds of unkeyed forward mixing, 8 rounds of keyed forwards transformation, 8 rounds of keyed backwards transformation, and 8 rounds of unkeyed backwards mixing and keyed subtraction. The rounds use +—x rotates XOR and two fixed 8-bit S-boxes.
RC6	RSA Laboratories, United States	A 20-round iterative cipher, developed from RC5 (and fully parameterized), which uses a number of 32-bit operations (+—x XOR rotates) to mix data in each round.
Rijndael	Joan Daemen and Vincent Rijmen, Belgium	A 10- to 14-round iterative cipher, using byte substitution, row shifting, column mixing, and key addition, as well as an initial and final round of key addition, derived from the previous SQUARE cipher.
SAFER+	Cylink Corp., United States	An 8- to 16-round iterative cipher, derived from the earlier SAFER cipher. SAFER+ uses +x XOR and two fixed 8-bit S-boxes.
SERPENT	Ross Anderson (U.K.), Eli Biham (Israel), and Lars Knudsen (Norway)	A 32-round Feistel cipher, with key mixing using XOR and rotates, substitutions using 8 key-dependent 4-bit S-boxes, and a linear transformation in each round.
Twofish	Bruce Schneier et al., United States	A 16-round Feistel cipher using four key-dependent 8-bit S-boxes, matrix transforms, rotations, and based in part on the Blowfish cipher.

[a] From http://www.adfa.edu.au/~lpb/papers/unz99.html.

EXHIBIT 89.2 Five Algorithms Selected by NIST

Algorithm	Main Strength	Main Weaknesses
MARS	High security margin	Complex implementation
RC6	Very simple	Lower security margin as it used operations specific to 32-bit processors
Rijndael	Simple elegant design	Insufficient rounds
Serpent	High security margin	Complex design and analysis, poor performance
Twofish	Reasonable performance, high security margin	Complex design

AES implementations have already emerged in many commercial software security products as an optional algorithm (in addition to Triple-DES and others). Software implementations have always come before hardware products due to the inherent time it takes to design and update hardware. It is generally easier to upgrade software than to perform a hardware replacement or upgrade, and many vendors have already incorporated AES into their latest designs.

For those organizations already running Triple-DES, there are not many compelling reasons (except for compatibility) to immediately use AES. It is likely that the speed at which companies upgrade to AES will increase as more products ship in AES-enabled mode.

89.3 Rijndael

Rijndael, the AES candidate, was developed by Dr. Joan Daemen of Proton World International and Dr. Vincent Rijmen, a postdoctoral researcher in the electrical engineering department of Katholieke Universiteit of the Netherlands.[12] Drs. Daemen and Rijmen are well-known and respected in the cryptography community. Rijndael has its roots in the SQUARE cipher,[13] also designed by Daemen and Rijmen.

The details on Rijndael are specified in its original AES proposal.[14] From a technical perspective,[15] Rijndael is a substitution-linear transformation network (i.e., non-Feistel[16,17]) with multiple rounds, depending on the key size. Rijndael's key length and block size is either 128, 192, or 256 bits. It does not support arbitrary sizes, and its key and block size must be one of the three lengths.

Rijndael uses a single S-box that acts on a byte input in order to give a byte output. For implementation purposes, it can be regarded as a lookup table of 256 bytes. Rijndael is defined by the equation

$$S(x) = M(1/x) + b$$

over the field GF(2^8), where M is a matrix and b is a constant.

A data block to be processed under Rijndael is partitioned into an array of bytes and each of the cipher operations is byte oriented. Rijndael's ten rounds each perform four operations. In the first layer, an 8×8 S-box (S-boxes used as nonlinear components) is applied to each byte. The second and third layers are

[12] For a quick technical overview of Rijnadel, see http://www.baltimore.com/devzone/aes/tech_overview.html.

[13] www.esat.kuleuven.ac.be/~rijmen/square/index.html.

[14] Available at www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaeldocV2.zip.

[15] http://csrc.nist.gov/encryption/aes/round2/r2report.pdf.

[16] Feistel ciphers are block ciphers in which the input is split in half. Feistel ciphers are provably invertible. Decryption is the algorithm in reverse, with subkeys used in the opposite order.

[17] Of the four other AES finalists, MARS uses an extended Feistel network; RC6 and Twofish use a standard Feistel network; and Serpent uses a single substitution-permutation network.

linear mixing layers, in which the rows of the array are shifted and the columns are mixed. In the fourth layer, subkey bytes are XORed into each byte of the array. In the last round, the column mixing is omitted.[18]

89.4 Why Did NIST Select the Rijndael Algorithm?

According to the NIST,[19] Rijndael was selected due to its combination of security, performance, efficiency, ease of implementation, and flexibility.[20] Specifically, NIST felt that Rijndael was appropriate for the following reasons:

- Good performance in both hardware and software across a wide range of computing environments
- Good performance in both feedback and nonfeedback modes
- Key setup time is excellent
- Key agility is good
- Very low memory requirements
- Easy to defend against power and timing attacks (this defense can be provided without significantly impacting performance).

89.5 Problems with Rijndael

Although the general consensus is that Rijndael is a fundamentally first-rate algorithm, it is not without opposing views.[21] One issue was with its underlying architecture; some opined that its internal mathematics were simple, almost to the point of being rudimentary. If Rijndael were written down as a mathematical formula, it would look much simpler than any other AES candidate. Another critique was that Rijndael avoids any kind of obfuscation technique to hide its encryption mechanism from adversaries.[22] Finally, it was pointed out that encryption and decryption use different S-boxes, as opposed to DES which uses the same S-boxes for both operations. This means that an implementation of Rijndael that both encrypts and decrypts is twice as large as an implementation that only does one operation, which may be inconvenient on constrained devices.

The Rijndael team defended its design by pointing out that the simpler mathematics made Rijndael easier to implement in embedded hardware. The team also argued that obfuscation was not needed. This, in turn, led to speculation that the Rijndael team avoided obfuscation to evade scrutiny from Hitachi, which had expressed its intentions to seek legal action against anyone threatening its U.S.-held patents. Hitachi claimed to hold exclusive patents on several encryption obfuscation techniques, and had not been

[18]Known as the key schedule, the Rijndael key (which is from 128 to 256 bits) is fed into the key schedule. This key schedule is used to generate the sub-keys, which are the keys used for each round. Each sub-key is as long as the block being enciphered, and thus, if 128 bits long, is made up of 16 bytes. A good explanation of the Rijndael key schedule can be found at http://home.ecn.ab.ca/~jsavard/crypto/co040801.htm.

[19]http://csrc.nist.gov/encryption/aes.

20As clarified in the report by NIST (*Report on the Development of the Advanced Encryption Standard*), the fact that NIST rejected MARS, RC6, Serpent, and Twofish does not mean that they were inadequate for independent use. Rather, the sum of all benefits dictated that Rijndael was the best candidate for the AES. The report concludes that "all five algorithms appear to have adequate security for the AES."

[21]Improved Cryptanalysis of Rijndael, N. Ferguson, J. Kelsey, et al., www.counterpane.com/rijndael.html.

[22]Contrast this with Twofish; see *The Twofish Team's Final Comments on AES Selection*, www.counterpane.com/twofish-final.html.

forthcoming about whether it would consider licensing those techniques to any outside party.[23] In fact, in early 2000, Hitachi issued patent claims against four of the AES candidates (MARS, RC6, Serpent, and Twofish).

89.6 Can AES Be Cracked?

Although a public-DES cracker has been built[24] as detailed in Cracking *DES: Secrets of Encryption Research, Wiretap Politics and Chip Design*, there still exists the question of whether an AES-cracking device can be built?

It should be noted that after nearly 30 years of research, no easy attack against DES has been discovered. The only feasible attack against DES is a brute-force exhaustive search of the entire keyspace. Had the original keyspace of DES been increased, it is unlikely that the AES process would have been undertaken.

DES-cracking machines were built that could recover a DES key after a number of hours by trying all possible key values. Although an AES cracking machine could also be built, the time that would be required to extricate a single key would be overwhelming.

As an example, although the entire DES keyspace can feasibly be cracked in less than 48 hours, this is not the case with AES. If a special-purpose chip, such as a field-programmable gate array[25] (FPGA), could perform a billion AES decryptions per second, and the cracking host had a billion chips running in parallel, it would still require an infeasible amount of time to recover the key. Even if it was assumed that one could build a machine that could recover a DES key in a second (i.e., try 2^{55} keys per second), it would take that machine over 140 trillion years to crack a 128-bit AES key.

Given the impenetrability of AES (at least with current computing and mathematical capabilities), it appears that AES will fulfill its requirement of being secure until 2030. But then again, a similar thought was assumed for DES when it was first designed.

Finally, should quantum computing transform itself from the laboratory to the realm of practical application, it could potentially undermine the security afforded by AES and other cryptosystems.

89.7 The Impact of AES

The two main bodies to put AES into production will be the U.S. government and financial services companies. For both entities, the rollout of AES will likely be quite different.

For the U.S. government sector, after AES is confirmed as a FIPS, all government agencies will be required to use AES for secure (but unclassified) systems. Because the government has implemented DES and Triple-DES in tens of thousands of systems, the time and cost constraints for the upgrade to AES will be huge.

AES will require a tremendous investment of time and resources to replace DES, Triple-DES, and other encryption schemes in the current government infrastructure. A compounding factor that can potentially slow down the acceptance of AES is the fact that because Triple-DES is fundamentally secure (its main caveat is its speed), there is no compelling security urgency to replace it. Although AES may be required, it may be easier for government agencies to apply for a waiver for AES as opposed to actually implementing it.[26] With the budget and time constraints of interchanging AES, its transition will occur over time, with economics having a large part in it.

[23] www.planetit.com/techcenters/docs/security/qa/PIT20001106S0015.

[24] It is an acceptable assumption to believe that the NSA has had this capability for a long time.

[25] An FPGA is an integrated circuit that can be programmed in the field after manufacture. They are heavily used by engineers in the design of specialized integrated circuits that can later be produced in large quantities for distribution to computer manufacturers and end users.

[26] Similar to those government agencies that applied for waivers to get out of the requirement for C2 (*Orange Book*) certification.

The financial services community also has a huge investment in Triple-DES. Because there is currently no specific mandate for AES use in the financial services community, and given the preponderance of Triple-DES, it is doubtful that any of the banking standards bodies will require AES use.

While the use of single DES (also standardized as X9.23-1995, Encryption of Wholesale Financial Messages) is being withdrawn by the X9 committee (see X9 TG-25-1999); this nonetheless allows continued use of DES until another algorithm is implemented.

But although the main advantages of AES are its efficiency and performance for both hardware and software implementations, it may find a difficult time being implemented in large-scale nongovernmental sites, given the economic constraints of upgrading it, combined with the usefulness of Triple-DES. Either way, it will likely be a number of years before there is widespread use of the algorithm.

For Further Information

1. Savard, John, How Does Rijndael Work? www.securityportal.com/articles/rijndael20001012.html and http://home.ecn.ab.ca/~jsavard/crypto/co040801.htm.
2. Tsai, Melvin, AES: An Overview of the Rijndael Encryption Algorithm, www.gigascale.org/mescal/forum/65.html.
3. Landau, Susan, Communications Security for the Twenty-first Century: The Advanced Encryption Standard and Standing the Test of Time: The Data Encryption Standard, www.ams.org/notices/200004/fea-landau.pdf and www.ams.org/notices/200003/fea-landau.pdf.
4. Schneier, Bruce, *Applied Cryptography*, John Wiley & Sons, 1996.
5. Menezes, Alfred, *Handbook of Applied Cryptography*, CRC Press, 1996.
6. Anderson, Ross, *Security Engineering*, John Wiley & Sons, 2001.
7. Brown, Lawrie, A Current Perspective on Encryption Algorithms, http://www.adfa.edu.au/~lpb/papers/unz99.html.