Interested in learning more about security?

SANS Institute InfoSec Reading Room

Who's Who in AES?

There are several encryption methods in use today using various algorithms depending on the information being protected. Some are used to protect unclassified but sensitive data and other secret algorithms are used to protect the most highly classified data. This paper is going to introduce the new Advanced Encryption Standard, or AES, the winning algorithm, its competitors, the specifications set forth, and decision making process of NIST. The new planned Federal Information Processing Standard cryptographic

Kyle Jones
Version 1.2e

Who's Who in AES?

Encryption is an age-old method of scrambling information in such a way that only the originator and intended recipients have the ability to return the information back to its original format. Its main purpose is to prevent valuable information from falling into the wrong hands and possibly causing irreparable damage. An example of a simple type of encryption used against me involves a different language. A friend of mine is bilingual, speaking both English and Spanish fluently. When I visit his home and his family is present, he speaks Spanish to his family in certain conversations and English in others. Knowing that I don't speak or understand Spanish, the Spanish conversations are encrypted and that information remains private among the family. The main goal in designing any encryption algorithm is security. There are several encryption methods in use today using various algorithms depending on the information being protected. Some are used to protect unclassified but sensitive data and other secret algorithms are used to protect the most highly classified data. This paper is going to introduce the new Advanced Encryption Standard, or AES, the winning algorithm, its competitors, the specifications set forth, and decision making process of NIST.

The new planned Federal Information Processing Standard cryptographic algorithm to be implemented by the United States Government is AES, or Advanced Encryption Standard. The algorithm selected is Rijndael developed by Belgium cryptographers Dr. Joan Daemen and Dr. Vincent Rijmen. The Rijndael / AES marriage marks a four year effort in developing the next accepted standard of encryption to be implemented in government organizations. It will be used domestically and internationally in both the federal and private sectors. AES is set to replace DES, which has been in use for over twenty years and was cracked in less than 24 hours at a cracking contest in 1999. Single DES is also being eliminated and is currently only used in Legacy systems. The one algorithm that's not expected to be phased out anytime soon is Triple DES.

Triple DES uses DES to encrypt data three times. The first key encrypts data the first time; the second key encrypts the data a second time, and the first key is used again to encrypt the data a third time. Though Triple DES is expected to remain in use for some time, problems are clearly visible for the future. Having to encrypt data three times before transmitting dominates CPU resources. With the five AES finalists all having much more efficient operations and flexible implementation options, its expected that Triple DES will be around, it just won't be used that much. The differences between AES and Triple DES aren't about strength of security, but rather about superior performance and resource allocation.

A request came out of the U.S. Commerce Department by NIST in 1997 for a new algorithm to bump DES. When NIST announced the need for AES, they also set the specifications the new standard should encompass. AES will be an unclassified, public, royalty free worldwide encryption algorithm. It will be unclassified and available to the public to maximize public scrutiny and comment in an effort to provide the best analysis of the algorithms. After all, the global cryptographic community will provide the best input being that they're going to be the ones trying to break the AES submissions. The government announcing the need to develop a

new standard encryption algorithm in an open atmosphere cooperating and participating with the private sector was one of its finest moments. The algorithm will use block encryption, and at a minimum support key sizes of 128, 192, and 256 bits with a 128 bit block size. It must offer a security level that will protect data for 20 to 30 years. Each algorithm will be tested heavily against all known linear and differential cryptanalysis attacks. Efforts will even be made to ensure that it will withstand unknown attacks. A new development in code cracking research finds that the algorithms could possibly be cracked by an attack involving power consumption. By monitoring the power consumption of smart cards, it's possible for researchers to crack the code. Research into a solution for the attack is the development of hardware that is able to change its power consumption signature.

It's impossible to know just how long an algorithm will last. With so much emphasis being put on security, the flip side of that coin is the heavy emphasis put on breaking the security of the algorithm. Sure, the competition tried to crack the other submitted algorithms, but that cracking was monitored. What about the cracking attempts that aren't monitored? Is it possible that another country has cracked Triple DES or even one of the new submitted AES algorithms? Of course it is. DES has been around since the seventies. Technology has changed so much, providing capabilities today that couldn't even be imagined in the seventies. It's close-minded to think that AES will have the same lifespan as DES. Some researchers are expecting a lifespan of only five years. Part of the controversy that NIST encountered when writing the AES guidelines dealt with the need to have more than one algorithm for AES, and having one or more backup algorithms.

NIST believes that top performance is achieved through a hardware implementation so each candidate must be efficient in silicon. The algorithm must be successful on Intel Pentium processors because the encryption needs to work on non-traditional devices. Success across all Pentium platforms is critical since Pentium dominates the desktop computing environment today. Finally, it will operate easily and efficiently on 8 bit CPU smart cards. 32 bit CPU smart cards will be tested to provide researchers insight into the future. 8 bit CPU's are the norm today but the 32 bit CPU is on the horizon and will one day be the norm.

After two years of competition among fifteen algorithms, the field was narrowed to five finalists. In 1999, NIST selected MARS from IBM Research, RC6 from RSA Labs, Serpent, Twofish, and Rijndael. The formulas were tested extensively and measured against metrics of encryption/decryption speeds, attack resistance, and key/algorithm set up time. On October 2, 2000, it was announced that Rijndael had won out the competition and became the sole candidate for AES.

Rijndael is a simple self-supporting cipher that does not depend on or use other cryptographic components. It was designed based on the following three principles: it must defend against both known and unknown attacks, code density and speed across a variety of platforms, and simplicity. It uses a fast block cipher square algorithm that can be efficiently implemented across a variety of platforms. The Rijndael cipher design uses a variable number of rounds. It has nine rounds if 128 bit length is used for the block and key. It has eleven rounds if a 192 bit length is used for either the block or key. If bit length is 256 for either the block or key, there are thirteen rounds.

Rijndael received a high mark in its performance on 32 bit CPU platforms. Having a different algorithm for encryption and decryption, the performance numbers were nearly identical when both were implemented. Smart card performance reviews proved Rijndael to have one of the fastest speeds of the finalists with a RAM footprint of 36 bytes for encryption only. Adding the

decryption function slightly raises RAM requirements but still leaves Rijndael very competitive. It also had an impressive showing in a hardware implementation. The on-the-fly key schedule makes for high key agility, showing key change measurements as taking zero time. The lowest mark received by Rijndael was due to it having separate operations for encryption and decryption in the building blocks.

We've all heard about exploits and weaknesses that exist for Windows NT, IIS, etc., and encryption is no different. Cryptographers, both good and bad, work around the clock to design tools that will crack encryption algorithms in an effort to get their hands on valuable information that may return a high monetary value. An attack does exist that can break six rounds of Rijndael, but developers state that six rounds is far from a threatening exploit and it poses no problems for the future.

MARS was developed by IBM and is an acronym for Multiplication, Addition, Rotation, and Substitution, which are terms that describe how the algorithm operates. MARS was designed with two main objectives in mind: security and performance. It was built to offer the highest security standard yet have stellar performance as a symmetric block cipher. IBM even goes so far to state that MARS is more secure than their very own 3DES, and in the face of today's cryptanalytical attacks, is impossible to crack. Cryptanalytical attacks usually focus on stripping off the top and bottom rounds to mount the attack on the middle or core rounds. MARS was designed with a mixed structure, designing the top and bottom rounds very differently than the middle rounds increasing its resistance to new and undiscovered attacks.

MARS is a stellar performer using a dedicated hardware solution. There is a 10X increase in performance over a software based implementation. There are tradeoffs in both scenarios, that's why MARS was designed with the flexibility needed for the varying computer environments of today. MARS can achieve rates of 100mbps on todays high-end computers, approximately 65mbps on a Pentium Pro 200, and 85mbps on a Power PC 200, making it the fastest in a C version of the AES finalists. Performance numbers in assembler tell a different story. Here, the algorithm's encryption round has a lengthy functional path and core processing time is significantly increased on a "clocks per block" basis. For this reason, it's considerably slower on a Pentium platform. MARS received a poor grade in smart card performance. It requires 2k bytes of ROM for the S-box function only, which is more than the total ROM used by other finalists. The RAM requirement is about 200 bytes since MARS is not designed with on-the-fly subkey generation. The IBM design team states that MARS smart card implementation is based on smart cards with dedicated crypto units.

Having four different round functions, the MARS cryptographic strength lies in its core. Attacks on the core mounted eleven of the sixteen total rounds with one cipher attack symmetrically reducing four different round functions from eight to three.

Serpent is the candidate for AES submitted as a global endeavor by Cambridge University, University of Haifa, Israel, and the University of Bergen, Norway. It's simply an improved version of DES. Serpent received 59 votes at the last AES conference, which was second to the winning Rijndael that received 86 votes. Serpent and Rijndael are similar with the main difference being speed: Rijndael is faster having fewer rounds than Serpent. Serpent is a 32 bit round cipher using four 32 bit data blocks giving it a 128 bit block size. The 32 rounds encrypt a 128 bit plaintext to a 128 bit ciphertext using thirty-three 128 bit subkeys.

Serpent's performance is achieved through the efficient use of parallelism with its speed being independent of key length. Serpent's key setup and block encryption time is the same regardless if the key is 128, 192, or 256 bits in length. Analyses of the five finalists in a software

implementation using both C and Assembly languages proved that Serpent had the slowest encryption speed using 128 bit keys. Serpent's best software performance is three to four times slower than the other finalists. Even a conservative sixteen round Serpent implementation would be the slowest. The algorithm uses RISC operations, which suggest its normal performance will not be harmfully affected across all test platforms. Serpent's smart card test put the algorithm into the category of "algorithms that can fit on any smart card," using less than 128 bytes of RAM. Its on-the-fly key schedule allows for minimal RAM requirements. Serpent received a high rating in a hardware solution due to its performance tradeoffs, key agility, and the speed of its round function.

The Serpent design resists all known attacks with wide safety margins in both differential and linear attack methods. Analysis shows no warning of any functional shortcut attack; such a discovery would be an academic breakthrough. In deciding on a lifespan for Serpent, the development team designed it to have a useful service lifetime plus an average human lifetime concluding that it will last at minimum a century.

RC6 is the brainchild of RSA Labs developed for the AES challenge. The RC6 architecture is based on the RC5 design, which was modified only to meet the AES specifications set forth by NIST. The key schedule for RC6 came from RC5, which is important because this key schedule has been studied for over six years. RSA designed RC6 to offer optimal security, performance, and simplicity. RSA's goal was to fulfill as many objectives as possible and still keep the algorithm simple. With only twelve lines to the algorithm, it ranked as the simplest of the finalists with Rijndael a near second.

Attacks on RC6 have mounted up to fifteen of its twenty rounds. Submission papers support this number by pointing out that it would be possible to mount sixteen rounds. Even with sixteen rounds mounted, RC6 offers an acceptable security margin needed for the future.

RC6 performance was tuned for optimization in a software implementation. RSA put a higher priority on a software rather than a hardware implementation because software implementations have become the norm. Using the Pentium and Power PC processors, RC6 outperformed Rijndael. The performance numbers between the two algorithms were comparable but there were instances in which RC6 outperformed Rijndael by a factor of two or more. Testing in a 32 bit environment seems to provide the realistic differences between the two algorithms, with Rijndael outperforming RC6 in most cases. When tested on advanced processors, Rijndael outperformed RC6 because it took advantage of parallelism where RC6 did not. Performance on the Pentium Pro and Pentium II proved RC6 to be the fastest algorithm in assembly language. It fared very well in C language but suffered considerably from the design of most C compilers. The big surprise came when RC6 was tested on a base Pentium MMX chip. It performed nearly three times slower than on the Pentium Pro, whereas the other finalists have nearly identical performance on the same platforms. Findings show that RC6 does not depend on basic RISC instructions whereas the other finalists do, making RC6 very inconsistent across the various platforms. It was also not among the fastest in smart card performance. Poor performance came on the 8 bit CPU because RC6 does not have a multiply instruction. Having no on-the-fly subkey generation and high RAM requirements, RC6 will no fit on most smart card CPU's.

There are areas in which Rijndael and RC6 proved equally suitable. There are other areas where RC6 proved to be an alternative to Rijndael, but overall performance proved Rijndael to be superior in several application areas.

Twofish is the AES candidate submitted by the security consulting firm Counterpane. Twofish originated from the original Blowfish design but was modified for a 128 bit data block. It's a

symmetric block cipher, which uses a single key for encryption and decryption. Twofish is a mean, lean, simple algorithm that doesn't use new security methodology or design elements.

Twofish flexibility received an excellent rating providing a variety of implementation options. Implementation flexibility comes from the performance tradeoff options that are built in the layered key schedule. It performs well on both high-end CPU's and tiny smart card CPU's. It can be set up for fast encryption, which means key setup takes longer. This works well when encrypting large amounts of plaintext data with the same key. If encryption speed isn't an issue, then it can be setup for a fast key setup, which works well when encrypting data blocks that change keys rapidly. The Counterpane team worked on a design that could be tuned for optimization in either a software or hardware implementation rather than offering optimization in one or the other. On the standard 6805 smart card processor, Twofish operated with less than 64 bytes of RAM having different space-time tradeoff options. A noted recommendation to further increase performance on smart card CPU's is to use those that have a second index register such as the 6502.

The Counterpane team spent over one thousand hours cryptanalyzing Twofish ensuring that it would be strong against known and possible unknown attacks. Several attacks were mounted against the algorithm with the best attack cracking 5 of the 16 rounds. Through lots of research and analysis, it was found that key differential attacks against a full Twofish implementation just would not crack the algorithm. Twofish has a key schedule designed to resist even the nastiest attacks.

Of the five finalists, Twofish had the most flexibility in implementation. It performed efficiently on a wide range of microprocessors including smart cards and dedicated hardware solutions. Twofish is free with no rules governing its use and users are encouraged to download Twofish for use in their applications.

Rijndael is currently available and there are a handful of products released that support it. Acceptance is expected to fall into two categories: startups looking to set the trend early for marketshare and name recognition, and established industry leaders. A few major companies that have released statements saying they will support and use AES are Checkpoint Software, Cisco Systems, and RSA Labs. Products with AES are expected to ship this fall to minimize risks involved with losing out on federal contracts. Wide implementation of AES is predicted in 2004.

IT executives will require AES on their networks for the same reasons it was developed: fast encryption and compatibility with a wide range of equipment. Not using AES will require several encryption technologies in place for specific applications such as wireless communications, financial operations, and quality of service. Specific industries such as healthcare, banking, and finance will rush the AES implementation into their networks to provide a better layer of security. In turn, they will use AES in their marketing strategy to sell better security to the customer and increase marketshare. With the celebrity status given to hacker groups, consumers today have to be more concerned than ever with the privacy and confidentiality of their medical and financial information. Why shouldn't they be? The news talks of defaced websites and credit card numbers being stolen by the thousands way too often.

No AES finalists had the best performance in all areas of the competition but Rijndael obviously offered something that the competition didn't. There's the belief that NIST picked the foreign Rijndael to promote and encourage international relations but even the other finalists discredit the rumor. It's hard to say whether Rijndael will withstand the next twenty to thirty years as the sole AES algorithm. With technology moving as fast as it is, Rijndael certainly has

an interesting road to travel ahead. With all the hard work that the various teams put forth, one can be sure that the next time an algorithm selection process is front and center, it will be amazing at the new developments and breakthroughs that have been achieved in order to be the best of the best. In the AES challenge, there was no room for second best, and one can be assured the other algorithm design teams are going to make sure that it doesn't happen again.

References

Anderson, Ross, Eli Biham, and Lars Knudsen. "Serpent: A Proposal for the Advanced Encryption Standard." University of Cambridge Computer Laboratory 2000. 2 Aug. 2001 < http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/serpent.pdf>.

Anderson, Ross, Eli Biham, and Lars Knudsen. "The Case for Serpent." University Of Cambridge Computer Laboratory 24 Mar. 2000. 1 Aug. 2001 <http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/serpentcase.pdf>.

Baran, Nick. "NIST Selects AES Algorithm." Dr. Dobbs Journal (2001). 1 Aug. 2001 < http://www.ddj.com/articles/2000/0065/0065j/0065j.htm?topic=security>.

Burwick et al., "MARS – A Candidate Cipher for AES." IBM Research 1999. 5 Aug. 2001 < http://www.research.ibm.com/security/mars.pdf>.

Daemen, Joan, Lars R. Knudsen, and Vincent Rijmen. "The Block Cipher Square Algorithm." Dr. Dobbs Journal (1997). 1 Aug. 2001 <http://www.ddj.com/articles/1997/9710/9710e/9710e.htm>.

IBM MARS Team. "MARS and the AES Selection Criteria." IBM Research 2000. 4 Aug. 2001 < http://www.research.ibm.com/security/final-comments.pdf>.

National Institute of Standards and Technology. Computer Security Division. ADVANCED ENCRYPTION STANDARD (AES) Questions and Answers. Jan. 2001. 1 Aug. 2001 < http://csrc.nist.gov/encryption/aes/aesfact.html>.

Reavis, Jim. "Advanced Encryption Standard – Crypto for the Next Century." Network World Fusion 27 Sept. 1999. 1 Aug. 2001 <http://www.nwfusion.com/newsletter/Sec/0927sec1.html>.

Reavis, Jim. "Feature: Goodbye DES, Hello AES." Network World Fusion 30 July 2001. 1 Aug. 2001 < http://www.nwfusion.com/research/2001/0730feat2.html>.

Rivest, Ronald L., M.J.B. Robshaw, and Yiqun Lisa Yin. "RC6 as the AES." RSA Laboratories 2000. 1 Aug. 2001 < ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6-Statement.pdf>.

Schneier, Bruce. "The Twofish Encryption Algorithm." Dr. Dobb's Journal (1998). 8 August 2001 <http://www.ddj.com/articles/1998/9812/9812b/9812b.htm>.

Seifried, Kurt. “Advanced Encryption Standard Released.” Security Portal 5 Oct. 2000. 1 Aug. 2000 < http://www.securityportal.com/articles/aes20001003.html>.

Syed, Furqan. “Children of DES – A Look at the Advanced Encryption Standard.” Europe’s Information Security Event 2001. 1 Aug. 2001 <http://www.infosec.co.uk /page.cfm/Link=169>.

Last Updated: September 22nd, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

Rocky Mountain Fall 2017	**Denver, COUS**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
SANS Baltimore Fall 2017	**Baltimore, MDUS**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
Data Breach Summit & Training	**Chicago, ILUS**	**Sep 25, 2017 - Oct 02, 2017**	**Live Event**
SANS Copenhagen 2017	**Copenhagen, DK**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
SANS London September 2017	**London, GB**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
SANS Oslo Autumn 2017	**Oslo, NO**	**Oct 02, 2017 - Oct 07, 2017**	**Live Event**
SANS DFIR Prague Summit & Training 2017	**Prague, CZ**	**Oct 02, 2017 - Oct 08, 2017**	**Live Event**
SANS Phoenix-Mesa 2017	**Mesa, AZUS**	**Oct 09, 2017 - Oct 14, 2017**	**Live Event**
SANS October Singapore 2017	**Singapore, SG**	**Oct 09, 2017 - Oct 28, 2017**	**Live Event**
Secure DevOps Summit & Training	**Denver, COUS**	**Oct 10, 2017 - Oct 17, 2017**	**Live Event**
SANS Tysons Corner Fall 2017	**McLean, VAUS**	**Oct 14, 2017 - Oct 21, 2017**	**Live Event**
SANS Brussels Autumn 2017	**Brussels, BE**	**Oct 16, 2017 - Oct 21, 2017**	**Live Event**
SANS Tokyo Autumn 2017	**Tokyo, JP**	**Oct 16, 2017 - Oct 28, 2017**	**Live Event**
SANS Berlin 2017	**Berlin, DE**	**Oct 23, 2017 - Oct 28, 2017**	**Live Event**
SANS Seattle 2017	**Seattle, WAUS**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SANS San Diego 2017	**San Diego, CAUS**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SANS Gulf Region 2017	**Dubai, AE**	**Nov 04, 2017 - Nov 16, 2017**	**Live Event**
SANS Miami 2017	**Miami, FLUS**	**Nov 06, 2017 - Nov 11, 2017**	**Live Event**
SANS Milan November 2017	**Milan, IT**	**Nov 06, 2017 - Nov 11, 2017**	**Live Event**
SANS Amsterdam 2017	**Amsterdam, NL**	**Nov 06, 2017 - Nov 11, 2017**	**Live Event**
SANS Paris November 2017	**Paris, FR**	**Nov 13, 2017 - Nov 18, 2017**	**Live Event**
Pen Test Hackfest Summit & Training 2017	**Bethesda, MDUS**	**Nov 13, 2017 - Nov 20, 2017**	**Live Event**
SANS Sydney 2017	**Sydney, AU**	**Nov 13, 2017 - Nov 25, 2017**	**Live Event**
SANS London November 2017	**London, GB**	**Nov 27, 2017 - Dec 02, 2017**	**Live Event**
SANS San Francisco Winter 2017	**San Francisco, CAUS**	**Nov 27, 2017 - Dec 02, 2017**	**Live Event**
SIEM & Tactical Analytics Summit & Training	**Scottsdale, AZUS**	**Nov 28, 2017 - Dec 05, 2017**	**Live Event**
SANS Khobar 2017	**Khobar, SA**	**Dec 02, 2017 - Dec 07, 2017**	**Live Event**
SANS Munich December 2017	**Munich, DE**	**Dec 04, 2017 - Dec 09, 2017**	**Live Event**
European Security Awareness Summit & Training 2017	**London, GB**	**Dec 04, 2017 - Dec 07, 2017**	**Live Event**
SANS Austin Winter 2017	**Austin, TXUS**	**Dec 04, 2017 - Dec 09, 2017**	**Live Event**
SANS Frankfurt 2017	**Frankfurt, DE**	**Dec 11, 2017 - Dec 16, 2017**	**Live Event**
SANS Bangalore 2017	**Bangalore, IN**	**Dec 11, 2017 - Dec 16, 2017**	**Live Event**
SANS SEC504 at Cyber Security Week 2017	**OnlineNL**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**