Crypto Corner
Editors: Peter Gutmann, pgut001@cs.auckland.ac.nz
David Naccache, david.naccache@ens.fr
Charles C. Palmer, ccpalmer@us.ibm.com

The First 10 Years of Advanced Encryption

JOAN DAEMEN
STMicro-electronics

VINCENT RIJMEN
Katholieke Universiteit Leuven

On 2 October 2000, after a three-year study period in which 15 block ciphers competed, the US National Institute of Standards and Technology (NIST) announced that the block cipher Rijndael would become the Advanced Encryption Standard (AES).[1,2] This announcement will soon be 10 years old, reason enough for a first retrospective.

Block ciphers belong to the field of symmetric cryptography, which has existed for thousands of years. The first modern design of a block cipher can be attributed to Claude Shannon, who proposed designing a block cipher $\mathcal{B}$ as a *product cipher*.[3] Such a cipher concatenates two keyed simple substitution ciphers S and T, separated by a publicly known mixing transformation M:

$$\mathcal{B} = T \circ M \circ S.$$

Researchers quickly generalized this construction to *iterated ciphers*:

$$\mathcal{B} = S_r \circ M \circ \ldots \circ M \circ S_2 \circ M \circ S_1.$$

These ciphers often derive the different S_i from one common transformation S, parameterized by a *round key* k_i. The sequence $M \circ S[k_i]$ is a *round* of the block cipher; r is the number of rounds.

Rijndael's Design

Rijndael is a *key-iterated cipher*:[2] $S[k_i](x) = S(x) + k_i$, with + denoting the addition over GF(256) (the Galois field of 256 elements).

Design Principles

Rijndael's design philosophy follows three principles.

Keep it simple. We achieved simplicity by maximizing the symmetry in the round transformation. This transformation consists of a few components that can all easily be described in terms of operations over the finite field GF(256). All the transformation's elements are necessary; removing one leads to a weak design.

Performance is important. We wanted to achieve high performance on a wide range of platforms because high performance is key to achieving wide acceptance. The importance of performance is illustrated by the many weak designs that people have deployed because they're afraid that using the standard will unacceptably decrease performance.

Use well-understood components. Rijndael uses substitutions based on finite-field inversion, which had been studied thoroughly long before the AES competition started.[4] (Rijndael uses the map $x \rightarrow x^{-1}, \forall x \neq 0$, and $0 \rightarrow 0$.) The mixing transformation is based on the theory of error-correcting codes.

Rijndael's Strong Points

The design's simplicity makes the algorithm easy to understand and implement efficiently. It also facilitates understanding the mechanisms that give the algorithm its high resistance against differential cryptanalysis and linear cryptanalysis, to date the most important general methods of cryptanalysis in symmetric cryptography. The bounds' proofs are elegant and easy to understand.

The simplicity and ease of analysis greatly assisted us in completing and submitting our design with the limited human resources that were available.

Rijndael's Disadvantages

Some people contest simplicity as a design rule. A still commonly heard opinion is that simple designs are more prone to catastrophic failure. The underlying reasoning goes as follows: "If new attack techniques are being developed, then a simple design that didn't anticipate these techniques will easily fail, whereas the extra components that originally had no apparent function in a complex design could make the difference." Sometimes critics even confuse ease of analysis with ease of cryptanalysis: "Any design that can be understood must be insecure." Any symmetric property is suspect. Even though *we* are convinced that these reasonings are inherently flawed, they're still the cause of statements that must be studied, analyzed, and refuted.

However, Rijndael also has some generally accepted shortcomings. First, the finite-field

Table 1. The 17 SHA-3 (Secure Hash Algorithm 3) submissions using Advanced Encryption Standard (AES) components or components inspired by AES.

AES component	SHA-3 submissions using it*
AES round transformation	Arirang, *Echo*, Lane, Lesamnta, Shamata, *Shavite-3*, Twister, and Vortex
AES substitution ($x \longrightarrow x^{-1}$)	Aurora, Cheetah, *Fugue*, *Grøstl*, Sgail, and Spectral Hash
AES-like diffusion	Aurora, Cheetah, *Grøstl*, *Luffa*, and Sarmal
AES-inspired diffusion	*Fugue*, *JH*, and Sgail

* The submissions in italics were accepted for round 2 of the competition.

operations appeal to mathematically oriented minds but can be a burden for programmers. For instance, they aren't supported by popular scripting languages. Second, the mapping from finite-field elements to bit strings uses a suboptimal basis, which makes the substitution more costly in hardware than is strictly necessary. The fact that encryption and decryption can't use the same hardware, as is the case with Feistel ciphers, can be seen as a violation of the design principle of simplicity.

Finally, the key schedule didn't receive the same amount of attention during design as the cipher's other components, and this shows. Its performance is suboptimal, and for AES-192 and AES-256 (AES with key lengths of 192 and 256 bits), it's not strong enough to resist related-key attacks. (We discuss this in more detail later.)

AES Acceptance

Many standards and commercially available products have adopted AES, and researchers are adopting its strategy to design hash functions and other cryptographic primitives.

The US

AES's original and official scope is to protect sensitive but not classified data of the US federal government. In June 2003, the US Committee on National Security Systems allowed using AES for classified and secret information. It also allows using AES-192 and AES-256 for top secret information.[5] NIST has certified more than 1,000 AES products.

An important sign of recognition came in 2004, when NIST finally withdrew the Data Encryption Standard (DES). However, NIST still approves the use of 3-key triple DES.[6]

International

AES has been included in International Organization for Standardization (ISO) standards, Internet Engineering Task Force (IETF) standards and requests for comments, and IEEE standards. The Third-Generation Partnership Project (3GPP) Milenage suite of algorithms is based on Rijndael. We can safely claim that all software IT security products that support more than one algorithm also support AES.

For hardware and smart cards, AES acceptance has taken longer. In particular, the financial sector still relies mainly on DES; for example, EMV v4.2 (2008) still uses single DES for the generation of message authentication codes (MACing) and 2-key triple DES for encryption.[7] However, in July 2010, the optional use of AES was officially introduced in EMV. AES's slow uptake in this sector is due to legacy hardware. The block length and the minimal key length of 128 bits result in a relatively high lower bound on compact hardware implementations' size. A notable exception is the inclusion of special AES instructions in Intel's Westmere processor (2010).

AES and SHA-3

After the publication of the attacks on SHA-1 in 2004,[8] NIST was encouraged to organize the SHA-3 competition "like the AES competition" to find a new standard for hash functions. (SHA stands for Secure Hash Algorithm.) Of the 51 submissions accepted for round 1 of the competition, 17 used AES components or components inspired by AES (see Table 1). Of the 14 submissions accepted for round 2, six still use AES elements.

AES Security

Already before the AES competition, NIST made it clear that the winner would not only have to resist practical attacks but also have provable security against academic cryptanalysis methods.

Statistical Attacks

In October 2000, academic attacks were known for weakened variants of AES. Research showed that AES-128 with the number of rounds reduced from 10 to 6, AES-192 with the rounds reduced from 12 to 7, and AES-256 with the rounds reduced from 14 to 7 had less than optimal security against chosen-plaintext attacks. (In such attacks, the adversary has access to ciphertexts corresponding to plaintexts of its choice. Its task is to recover the secret key.) These results were close to those known at the AES competition's start.

Between 2000 and 2008, researchers refined the bounds related to linear and differential cryptanalysis.[9] Also, the best attacks were improved with one round: academic attacks were found for seven rounds of AES-128 and eight rounds of AES-192 and AES-256.[10]

Algebraic Attacks

In parallel with statistical cryptanalysis methods, AES's first years saw several attempts to break it through *algebraic attacks*. The starting observation for such attacks is that AES's only nonlinear component leads to relatively simple equations:

$$y = x^{-1} \Leftrightarrow xy = 1,\ x^2y = x,\ xy^2 = y.$$

Because in GF(256) the squaring operation is linear, you can convert the equations on the right to quadratic relations over GF(2). The probability that a randomly selected permutation over GF(256) can be described by a quadratic relation, let alone by three different ones, is very small.

Starting in 2002, critics have claimed that this choice of nonlinear component was unfortunate and would be AES's downfall. Researchers published a series of attacks—Extended Linearization (XL), Extended Sparse Linearization (XSL), and so on—all based on linearization of quadratic equations. None of the attacks could break more than trivially weakened versions of AES.

The Rebound Attack and Related-Key Attacks

In 2008, the *rebound attack* illustrated that hash function designs based on the AES design principles don't automatically inherit the security level.[11] Indeed, AES's bounds rely on a secret key for their strength.

In 2009, Alex Biryukov and Dmitry Khovratovich published the first results of related-key attacks on full AES.[12] They described attacks with a complexity of 2^{100} against AES-256 and a complexity of 2^{176} against AES-192. The results point out a weakness in the AES key schedule. However, the related-key attack model is very strong. The adversary has access to ciphertexts corresponding to plaintexts of its choice, encrypted under the secret key K or under one or more related keys K_i. This attack computes K_i by applying a function f_i to K, where the adversary can choose f_i.

Clearly, this attack has no practical impact on AES. Even the theoretical impact is limited; for instance, a cipher that's a pseudorandom permutation can be weak in this attack model.

For the next 10 years, we expect that AES will proliferate further and will replace DES and 3DES in products, designs, and cryptography textbooks. Of course, there will also be—and necessarily so—a continuing analysis of its properties and potential weaknesses. □

References

1. *Specification for the Advanced Encryption Standard (AES)*, Federal Information Processing Standards (FIPS) Publication 197, US Nat'l Inst. Standards and Technology, 2001.
2. J. Daemen and V. Rijmen, *The Design of Rijndael: AES—the Advanced Encryption Standard*, Springer, 2002.
3. C. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical J.*, vol. 28, no. 4, 1949, pp. 656–715.
4. K. Nyberg, "Differentially Uniform Mappings for Cryptography," *Advances in Cryptology—Eurocrypt 1993*, LNCS 765, Springer, 1993, pp. 55–64.
5. *National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information*, CNSS Policy No. 15, Fact Sheet No. 1, US Nat'l Security Agency, June 2003; http://csrc.nist.gov/groups/STM/cmvp/documents/CNSS15FS.pdf.
6. E. Barker et al., *Recommendation for Key Management Part 1: General (Revised)*, NIST Special Publication 800-57, US Nat'l Inst. Standards and Technology, Mar. 2007.
7. *EMV Integrated Circuit Card Specifications for Payment Systems, Book 2: Security and Key Management*, ver. 4.2, EMVCo, June 2008.
8. X. Wang, Y.L. Yin, and H. Yu, "Finding Collisions in the Full SHA-1," *Advances in Cryptology—Crypto 2005*, LNCS 3621, Springer, 2005, pp. 17–36.
9. L. Keliher and J. Sui, "Exact Maximum Expected Differential and Linear Probability for 2-Round Advanced Encryption Standard (AES)," *IET Information Security*, vol. 1, no. 2, 2007, pp. 53–57.
10. W. Zhang, W. Wu, and D. Feng, "New Results on Impossible Differential Cryptanalysis of Reduced AES," *Information Security and Cryptology—Icisc 2007*, LNCS 4817, Springer, 2007, pp. 239–250.
11. F. Mendel et al., "The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl," *Fast Software Encryption*, LNCS 5665, Springer, 2009, pp. 260–276.
12. A. Biryukov and D. Khovratovich, "Related-Key Cryptanalysis of the Full AES-192 and AES-256," *Advances in Cryptology—Asiacrypt 2009*, LNCS 5912, Springer, 2009, pp. 1–18.

Joan Daemen *is a cryptographer at STMicroelectronics. Contact him at joan.daemen@st.com.*

Vincent Rijmen *is a professor in the Electrical Engineering Department of the University of Leuven (Katholieke Universiteit Leuven). At the Graz University of Technology, he leads the Krypto research group of the Institute for Applied Information Processing and Communications. Contact him at vincent.rijmen@esat.kuleuven.be.*