

Advanced Encryption Standard: Important Key Issues

Ms. Priti Puri*, Hemant Tolani**

Lingaya's Institute Of Management And Technology

Abstract: This paper has described the survey of one important algorithm, advanced Encryption Standard (AES) of cryptography. The basics of this algorithm are explained in this paper. Three important issues, key creation, encryption and decryption are compared for the five competitors of AES. It has shown graphically. The hardware implementation of Rijndael algorithm at different platforms is also compared in this paper. Lastly, software implementation issues are discussed for Rijndael.

Keywords: AES, S-box, Subbyte, RC6.

I. INTRODUCTION

Security often requires that data be kept safe from unauthorized access. And the best line of defense is physical security (placing the machine to be protected behind physical walls). However, physical security is not always an option (due to cost and/or efficiency considerations). Instead, most computers are interconnected with each other openly, thereby exposing them and the communication channels that they use. There was a need to introduce the term “*Cryptography*” which helps in finding when an intermittent breaks the algorithm which the sender uses for sending with the help of any means. This problem can be broken down into five requirements that must be addressed:

- A. Confidentiality: assuring that private data remains private.
- B Authentication: assuring the identity of all parties attempting access.
- C. Authorization: assuring that a certain party attempting to perform a function has the permissions to do so.
- D. Data Integrity: assuring that an object is not altered illegally.
- E. Non-Repudiation: assuring against a party denying a data or a communication that was initiated by them.

Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. Generally speaking, it uses an [12] cryptographic system to transform a plaintext into a cipher text, using most of the time a key. Cryptography is a very important domain in computer science with many applications. The most famous example of cryptography is certainly the *Enigma machine*, the legendary cipher machine used by the German Third Reich to encrypt their messages, whose security breach ultimately led to the defeat of their submarine force.

There are secure public-key ciphers also, like the famous and very secure *Rivest-Shamir-Adleman* (commonly called RSA) that uses a public key to encrypt a message and a secret key to decrypt it.

Cryptography can be categorized into two units:

- a. Symmetric cryptography
- b. Asymmetric cryptography

Introduction to the Advanced Encryption Standard:

The Advanced Encryption Standard [12], in the following referenced as AES, is the winner of the contest, held in 1997 by the US Government, after the *Data Encryption Standard* was found too weak because of its small key size and the technological advancements in processor power. Fifteen candidates were accepted in 1998 and based on public comments the pool was reduced to five finalists in 1999. In October 2000, one of these five algorithms was selected as the forthcoming standard: a slightly modified version of the Rijndael.

The Rijndael, whose name is based on the names of its two Belgian inventors, *Joan Daemen* and *Vincent Rijmen*, is a *Block cipher*, which means that it works on fixed-length group of bits, which are called *blocks*. It takes an input block of a certain size, usually 128, and produces a corresponding output block of the same size. The transformation requires a second input, which is the secret key. It is important to know that the secret key can be of any size (depending on the cipher used) and that AES uses three different key sizes: 128, 192 and 256 bits.

While AES supports only block sizes of 128 bits and key sizes of 128, 192 and 256 bits, the original Rijndael supports key and block sizes in any multiple of 32, with a minimum of 128 and a maximum of 256 bits.

AES is a *substitution-permutation network*, which is a series of mathematical operations that use substitutions (also called S-Box) and permutations (P-Boxes) and their careful definition implies that each output bit depends on every input bit.

Description of the Advanced Encryption Standard algorithm

AES [12] is an iterated block cipher with a fixed block size of 128 and a variable key length. The different transformations operate on the intermediate results, called *state*. The state is a rectangular array of bytes and since the

block size is 128 bits, which is 16 bytes, the rectangular array is of dimensions 4x4. (In the Rijndael version with variable block size, the row size is fixed to four and the number of columns varies. The number of columns is the block size divided by 32 and denoted Nb). The cipher key is similarly pictured as a rectangular array with four rows. The number of columns of the cipher key, denoted Nk, is equal to the key length divided by 32.

It is very *important* to know that the cipher input bytes are mapped onto the the state bytes in the order a0,0, a1,0, a2,0, a3,0, a0,1, a1,1, a2,1, a3,1 ... and the bytes of the cipher key are mapped onto the array in the order k0,0, k1,0, k2,0, k3,0, k0,1, k1,1, k2,1, k3,1 ... At the end of the cipher operation, the cipher output is extracted from the state by taking the state bytes in the same order. AES uses a variable number of rounds, which are fixed: A key of size 128 has 10 rounds. A key of size 192 has 12 rounds. A key of size 256 has 14 rounds.

During each round, the following operations are applied on the state:

- a. Sub Bytes: every byte in the state is replaced by another one, using the Rijndael S-Box.
- b. Shift Row: every row in the 4x4 array is shifted a certain amount to the left
- c. Mix Column: a linear transformation on the columns of the state
- d. Add Round Key: each byte of the state is combined with a round key, which is a different key for each round and derived from the Rijndael key schedule

II. OBSERVATIONS

- A. The cipher key is expanded into a larger key, which is later used for the actual operations.[1]
- B. The round Key is added to the state before starting the with loop.
- C. The Final Round () is the same as Round (), apart from missing the Mix Columns () operation.
- D. During each round, another part of the Expanded Key is used for the operations.
- E. The Expanded Key shall ALWAYS be derived from the Cipher Key and never be specified directly.

Review of all the Competitors of AES

October 2, 2000 - NIST announces that Rijndael has been selected as the proposed AES.[14]

The National Institute of Standards and Technology (NIST) have been working with industry and the cryptographic community to develop an Advanced Encryption Standard (AES). The overall goal is to develop a Federal Information Processing Standard (FIPS) that specifies an encryption algorithm(s) capable of protecting sensitive government information well into the next century. The algorithm(s) is

expected to be used by the U.S. Government and, on a voluntary basis, by the private sector.

On January 2, 1997, NIST announced the initiation of the AES development effort and made a formal call for algorithms on September 12, 1997. The call stipulated that the AES would specify an unclassified, publicly disclosed encryption algorithm(s), available royalty-free, worldwide. In addition, the algorithm(s) must implement symmetric key cryptography as a block cipher and (at a minimum) support block sizes of 128-bits and key sizes of 128-, 192-, and 256-bits.

On August 20, 1998, NIST announced a group of fifteen AES candidate algorithms at the First AES Candidate Conference (AES1). These algorithms had been submitted by members of the cryptographic community from around the world. At that conference and in a simultaneously published Federal Register notice, NIST solicited public comments on the candidates. A Second AES Candidate Conference (AES2) was held in March 1999 to discuss the results of the analysis conducted by the global cryptographic community on the candidate algorithms. The public comment period on the initial review of the algorithms closed on April 15, 1999. Using the analyses and comments received, NIST selected five algorithms from the fifteen.

The AES finalist candidate algorithms were MARS, RC6, Rijndael, Serpent, and Twofish, and NIST developed a Round 1 Report describing the selection of the finalists.

These finalist algorithms received further analysis during a second, more in-depth review period prior to the selection of the final algorithm(s) for the AES FIPS. Until May 15, 2000, NIST solicited public comments on the remaining algorithms. Comments and analysis were actively sought by NIST on any aspect of the candidate algorithms, including, - but not limited to, - the following topics: cryptanalysis, intellectual property, crosscutting analyses of all of the AES finalists, overall recommendations and implementation issues. An informal AES discussion forum was also provided by NIST for interested parties to discuss the AES finalists and relevant AES issues.

Near the end of Round 2, NIST sponsored the Third AES Candidate Conference (AES3) - an open, public forum for discussion of the analyses of the AES finalists.

After the close of the Round 2 public analysis period on May 15, 2000, NIST studied all available information in order to make a selection for the AES. On October 2, 2000, NIST announced that it has selected Rijndael to propose for the AES. A report, press release, and AES fact sheet are available with that information.

A previous NIST publication entitled "Report on the NIST Java™ AES Candidate Algorithm Analysis"[2] documents the first round analysis performed by NIST, using the Java Development Kit (JDK) Version 1.1.6. Only IBM has

submitted official modifications to their candidate (MARS) prior to the final round. Results of the first round analysis using the JDK1.1.6 are therefore still valid for the other four candidates. The revised version of MARS was tested under both JDK1.1.6 and JDK1.3, to ensure an accurate comparison of the modified algorithm's performance in both environments.

Performance data for 128, 192, and 256-bit key sizes are also included in the second round analysis. Following charts are described for key setup, encryption and decryption for five competitors [11]

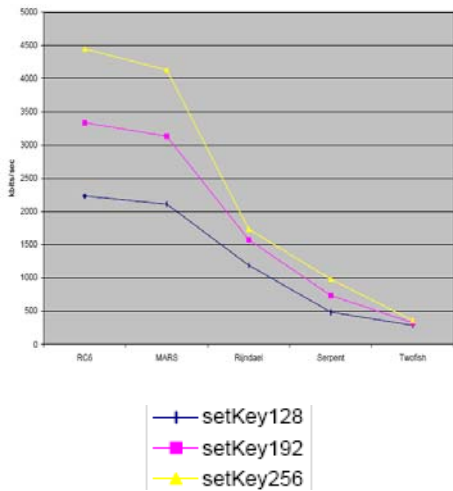


Chart 1: Key Setup

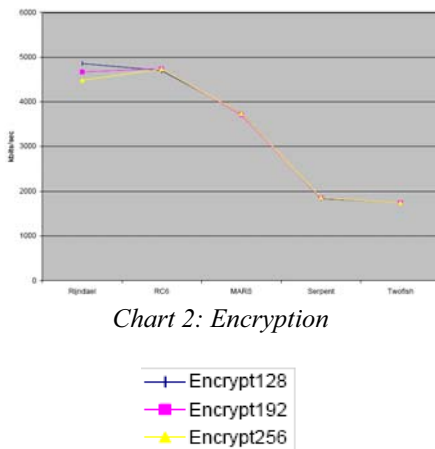


Chart 2: Encryption

The primary criteria used by NIST [10] to evaluate candidates for the new *Advanced Encryption Standard* (AES) include: security, efficiency in hardware and software, and flexibility. Among these four parameters, the efficiency in hardware appeared to be a particularly important factor used to differentiate among competing algorithms because

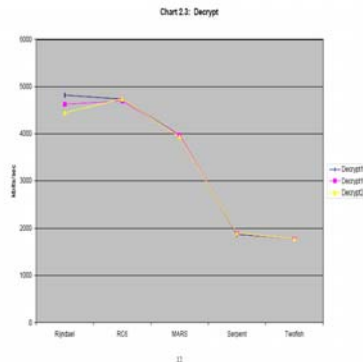


Chart 3: Decryption

- A. The comparison was based on a set of objective and commonly accepted measures.
- B. There existed large differences among AES candidates.
- C. There was a good agreement among results reported by several independent groups.

Cryptographic transformations can be implemented in both software and hardware.

Software implementations are designed and coded in programming languages, such as C, C++, Java, and assembly language, to be executed, among the other, on general-purpose microprocessors, digital signal processors, and smart cards. Hardware implementations are designed and coded in hardware description languages, such as VHDL and VerilogHDL, and are intended to be realized using two major implementation approaches:

- a. *Application Specific Integrated Circuits* (ASICs)
- b. *Field Programmable Gate Arrays* (FPGA). *Application Specific Integrated Circuits* (ASICs)

They are designed all the way from the behavioral description to the physical layout, and then sent for an expensive and time-consuming fabrication. *Field Programmable Gate Array* (FPGA) can be bought off the shelf and reconfigured by designers themselves. With each reconfiguration, which takes only a fraction of a second, an integrated circuit can perform a completely different

function. FPGA consists of thousands of universal building blocks, known as

Configurable Logic Blocks (CLBs), connected using programmable interconnects, as shown in Fig. 1a. Additionally, Virtex FPGAs contain dedicated memory blocks called

Embedded Array Blocks (EABs). Reconfiguration can change a function of each CLB and connections among them, leading to a functionally new digital circuit.

Hardware & Software Implementations of AES (Rijndael) in C/C++

- A. 6th December - improved the Visual Basic for Applications (VBA) example of how the AES DLL[15]

is used from VBA and added a Microsoft Word document that contains this VBA source code.

- B. 16th January 2002 - Significant simplification and restructuring of the code to make it easier to use. BUT NOTE that there are changes to the cipher interface. My thanks to David Hopwood for finding an error in the interface code and to Nigel Metheringham for some restructuring suggestions.
- C. 21st January - update to make code integration easier (suggested by Peter Gutmann).
- D. 28th January - update to correct an error in endian detection (my thanks to Gary Gorbet for testing this release on a wide range of big and little endian systems).
- E. 8th February - update to overcome macro expansion limits in some compilers when compiling aestab.c (reported by Peter Gutmann).

	ASICs (semi-custom and full-custom)	FPGAs	Software (general-purpose microprocessors)
<i>Performance characteristics</i>			
Parallel processing of data	yes	yes	limited
Pipelining	yes	yes	limited
Word size	variable	variable	fixed
Speed	very fast	fast	moderately fast
<i>Functionality</i>			
Algorithm agility	no	yes	yes
Tamper resistance	strong	limited	weak
Access control to keys	strong	moderate	weak
<i>Development process</i>			
Description languages	VHDL, Verilog HDL	VHDL, Verilog HDL	C, C++, Java, assembly language
Design cycle	long	moderately long	short
Design tools	very expensive	moderately expensive	inexpensive
Testing	expensive	moderately expensive	inexpensive
Maintenance and upgrades	expensive	inexpensive	inexpensive

- F. 5th April 2002 - minor release to remove a few TAB characters that confused some compilers and to add project files for Visual Studio .Net development.
- G. 8th June 2002 - a small update to remove a compilation failure that occurred on a rare combination of options (reported by Atasu Kubilay).
- H. 27 September 2002 - this release does not involve any bug fixes. It removes the use of the 'standard' fixed width integer types (more trouble than they are worth at the moment). It also changes the AES example code to use cipher text stealing.
- I. 6th June 2003 - added revised code for AES with support for royalty free combined encryption/authentication modes code implements both AES and Rijndael. The standard code implements block sizes of 16, 24 and 32 bytes, fixed during compilation, and a variable block size option covering these block sizes chosen at time of use. Each of these options operates with key sizes of 16, 24 and 32 bytes chosen at time of use. An alternative implementation offers block and key sizes of 16, 20, 24, 28 and 32 bytes. The standard implementation provides AES when implemented with a block size of 16 bytes.

This is heavily optimized, especially for the 16 byte key size. The variable block size option is much slower and is not recommended unless this is really needed. The alternative implementation is also less optimized.

CONCLUSION:

This paper has discussed the basics of Advanced Encryption Standard. It describes some information about the comparison of key creation, encryption and decryption for the five competitors of AES graphically. We also tried to show survey of hardware and software implementation of Rijndael algorithm in VHDL and C/C++.

REFERENCES

- [1] <http://www.progressive-coding.com/tutorial.php?id=0&print=1>
- [2] [AES] "Advanced Encryption Standard Development Effort," <http://www.nist.gov/aes>.
- [3] [AES3] "Third AES Candidate Conference,"
- [4] <http://csrc.nist.gov/encryption/aes/round2/conf3/aes3conf.htm>.
- [5] [DPR00b] A. Dandalis, V. K. Prasanna, J. D. Rolim, "A Comparative Study of Performance of AES Final Candidates Using FPGAs," Proc. Cryptographic Hardware and Embedded Systems Workshop, CHES 2000, Worcester, MA, Aug 17-18, 2000.
- [6] <http://ece.wpi.edu/Research/crypt/publications/index.html>.
- [7] <http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html>
- [8] J. Dray, "Report on the NIST Java™ AES Candidate Algorithm Analysis", <http://csrc.nist.gov/encryption/aes/round1/r1-java.pdf>, November 8, 1999.
- [9] A. Folmsbee, "AES Java™ Technology Comparisons", Proceedings of the Second Advanced Encryption Standard Candidate Conference, March 22, 1999, Pages 35-50.
- [10] http://ece.gmu.edu/crypto/AES_survey.pdf
- [11] <http://citeseer.ist.psu.edu/281667.html>
- [12] <http://www.progressive-coding.com/tutorial.php?id=0&print=1>
- [13] www.netsec.org.sa/cryptography.htm
- [14] http://www.et.informatik.uni-tuebingen.de/fileadmin/RI/teaching/netzsicherheit/ws0506/pdf/03_SymmetricCrypto_2on1.pdf
- [15] <http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf>