Third 2008 International Conference on Convergence and Hybrid Information Technology

A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography

Abdullah Al Hasib and Abul Ahsan Md. Mahmudul Haque
Helsinki University of Technology
Telecommunication Software and Multimedia Laboratory
Finland
hasib_iut@yahoo.com, ahaque@cc.hut.fi

Abstract

Security is always a major concern in the field of communication. Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithms are the two popular encryption schemes that guarantee confidentiality and authenticity over an insecure communication channel. There has been trifling cryptanalytic progress against these two algorithms since their advent. This paper presents the fundamental mathematics behind the AES and RSA algorithm along with a brief description of some cryptographic primitives that are commonly used in the field of communication security. It also includes several computational issues as well as the analysis of AES and RSA security aspects against different kinds of attacks including the countermeasures against these attacks.

1. Introduction

The fundamental necessity in security is to hide information from irrelevant public or malicious attackers. This requirement has given birth to different kinds of cryptographic primitives including symmetric and asymmetric cryptography, hash functions, digital signatures, message authentication codes etc.

Symmetric cryptography: In symmetric encryption, a key is shared between the sender and the receiver which is kept secret from the intruder. Among the different kinds of symmetric algorithms, Advanced Encryption Standard (AES) is gaining popularity due to its better security and efficiency than its predecessors [10]. It was defined in Federal Information Processing Standard (FIPS) 192, published in November 2001 [9, 8].

As a symmetric cipher, AES shares a secret key to encrypt and decrypt any message and operates on 128 bit fixed block. AES may be configured to use 3 different key-lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key.

Asymmetric cryptography: Unlike the symmetric cryptography, asymmetric cryptography uses a pair of keys to encrypt and decrypt message. One of these two keys is known as public key as it is distributed to others and the other is called private key which is kept secret. Normally public key is used to encrypt any message which can only be decrypted by the corresponding private key. There are essential properties that must be satisfied by the asymmetric cryptography [15]

- The key generation process should be computationally efficient.
- Sender should be able to compute the cipher text by using the public key of the receiver for any message.
- The receiver should be able to decrypt the cipher easily to plain text by using his own private key.
- It is impossible or at least impractical to compute the private key from the corresponding public key.
- It is computationally infeasible to compute the plain text form the public key and cipher text.

RSA is the most widely used asymmetric encryption system which was invented by Ronald Rivest, Adi Shamir, and Len Adleman in the year 1977 [4]. As a public key encryption standard, the private key is kept secret but the public key is revealed to everybody in RSA. Since its innovation, RSA is regarded as one of the most secure cryptosystems in existence.

The purpose of this paper is to describe the basic encryption and decryption method as well as to cover mathematical and security aspects of the two most widely used encryption schemes.

978-0-7695-3407-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICCIT.2008.179

The remainder of this paper is organized as follows. Section 2 contains a brief description on AES algorithm. In Section 3 AES computational issues have been described which is followed by the security efficiency in section 4. In section 5, RSA algorithm is elaborated. Section 6 and section 7 contain the computational and security aspects of RSA respectively. Section 8 contains the discussion and finally, the paper is ended with the conclusion at section 9.

2. Overview of AES Cipher

AES is an iterated cipher which was proposed by Joan Daemen and Vincent Rijmen (Rijndael) [8]. The proposed algorithm could support variable length block and key sizes e.g. multiple of 32 bits. However, only the 128 bit block size and 128, 192 and 256 bits keys are specified as AES standard. One of the key features of AES is that its structure is not based on Feistel network like its predecessor DES in which half of the data block is used to modify the other half of the data block and then the halves are swapped. Rather AES is based on S-P network in which the entire 128 bits input block is organized as 4x4 bytes array called State and is processed in several rounds. Number of rounds to be used depend on the length of key e.g. 10 round for 128 bit key, 12 rounds for 192-bit key and 14 rounds for 256 bit keys. Both in encryption and decryption process, the State array is modified at each round by a round function that defines four different byte-oriented transformations [1]:

1. *SubBytes transformation:* a non-linear substitution step where each byte is replaced with another byte according to a substitution table (S-box). The S-box is invertible and consists of the following two operations:
   - Inversion in the GF(2^8) field, modulo the irreducible polynomial m(x) = x^8 + x^4 + x^3 + x + 1.
   - Affine transformation defined as: Y = AX^{-1} + B, where A is a 8×8 fixed matrix and B is a 8×1 Vector

2. *ShiftRows transformation:* a transposition step where each row of the state is shifted cyclically a certain number of steps. This transformation can be defined as follows: $S^{'}_{r,c}$ = $S_{r,((c+shift(r,4))mod4)}$ where shift value shift(r,4) depends on the row number are as follows: shift(1,4) = 1; shift(2,4) = 2; shift(3,4) = 3;

3. *MixColumns:* a mixing operation which operates on the columns of the state, combining the four bytes in each column. The transformation can be written as the following matrix multiplication formula:

$$\begin{bmatrix} S^{'}_{0,c} \\ S^{'}_{1,c} \\ S^{'}_{2,c} \\ S^{'}_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad for\ 0 \leq c < 4$$

4. *AddRoundKey:* a simple bit wise XOR operation is performed between each byte of the state and the round key which is generated from the cipher key using Rijndael key schedule algorithm. The operation can be performed in the following way:

$$\left[\ S^{'}_{0,c} S^{'}_{1,c} S^{'}_{2,c} S^{'}_{3,c}\ \right] = \left[\ S_{0,c} \quad S_{1,c} \quad S_{2,c} \quad S_{3,c}\ \right] \oplus \left[\ W_{round*r+c}\ \right] for\ 0 \leq c < 4.$$

2.1. Encryption and Decryption process

The encryption and decryption process composed of several rounds depending on the size of the cipher key where each round performs some specific functions. In this paper, we have considered encryption and decryption process for 128 bit cipher key that requires 10 different rounds to complete the process. Fig. 1 depicts the steps involved in AES-128 algorithm

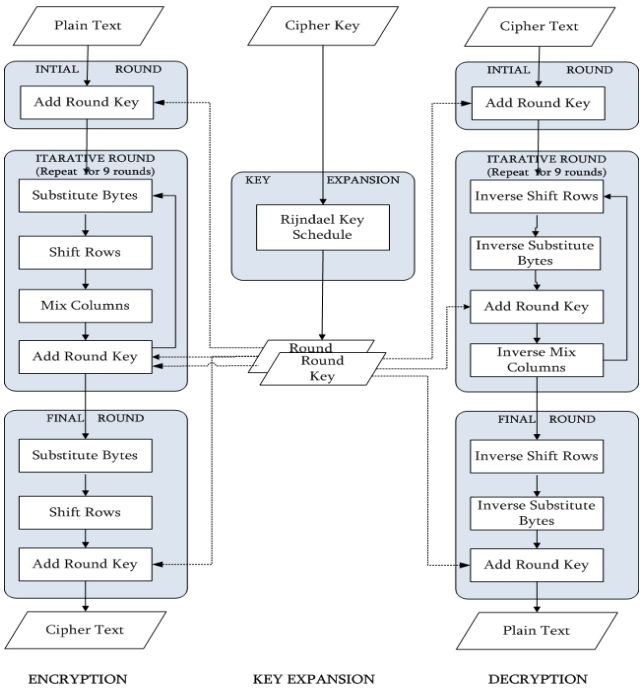

Figure 1. AES-128 algorithm

Both encryption and decryption process begins with the initial round i.e. round0 performs only the AddRoundKey transformation on the state arrray and provides the security as this is the only stage that makes use of the se-

cret key. Initial round is followed by nine identical rounds where each round incorporates SubBytes, Shiftrows, MixColumns and AddRoundKey transformations respectively on the state array. The final round is slightly different from the other rounds as it uses only three functions other than MixColumns transformation.

3 Computational issues of AES

In AES, each byte is considered as an element of Finite field of characteristic 2 with 8 terms, which is also known as Galois field, GF(2^8). GF(2^8) is actually a field of 256 elements and each element can be represented by a polynomial of degree 7 with coefficient $\{0, 1\}$.

Addition and Subtraction: The addition of two elements is performed by the bitwise XOR operation of the coefficients for the corresponding powers in the polynomials for the two elements.

Example: $(x^7 + x^5 + x^3 + x + 1) + (x^6 + x^5 + x + 1)$
$= (x^7 + x^6 + x^3)$

Multiplication: The multiplication can be obtained by performing ordinary polynomial multiplication and then taking the remainder of it by an irreducible polynomial of degree 8, $m(x) = (x^8 + x^4 + x^3 + x + 1)$

Example: $(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1)$
$=(x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1)$
$=(x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1)$
modulo $(x^8 + x^4 + x^3 + x + 1)$
$=(x^7 + x^6 + 1)$

4 Security of AES

No major security attack has been proven successful against the AES till now. The security strength of AES is described briefly against different kinds of attacks.

4.1 Brute Force Attack

The minimum length of AES cipher key is 128 bits that provides 2^{128} possible keys. Performing exhaustive search in this huge key space is considered infeasible. Thus the brute-force attack against AES with current and projected technology is considered impractical [3].

4.2 Mathematical Attack

AES uses S-box substitution table which is generated by determining the multiplicative inverse for a given number in Galois finite field and has the capability to resist the linear and differential cryptanalysis. Recently Warren D. Smith has defined an analytical method and claimed that there is a possibility of existence of a cracking algorithm that will be able to extract the AES 256 bits key from any random plaintext-ciphertext pairs [14]. However, it was just an assumption and he hasn't provided any cracking algorithm. As a result, till now, AES algorithm doesn't have any mathematical property that can be exploited by an attacker to reduce the effective key length and to gain success against AES.

4.3 Timing Attack

However, AES is less secure against side channels attacks. A number of side channel attacks have been proven successful including timing attack, power consumption, electromagnetic radiation, cache-collision timing attacks [6] etc. Timing attack is a ciphertext-only attack which is actually an implementation level attack that may occur in any implementation that doesn't run in fixed time, rather depends on the input. The high speed software implementation of AES is susceptible to this kind of attacks as it performs a sequence of S-box lookups that take variable time and dependent on the key. Daniel Bernstein has presented an example of such attack against a server running the OpenSSL AES implementation [5].

By selecting the cryptographic primitives which will allow efficient constant-time implementation we can avoid timing attack. However, it is extremely difficult to develop such primitives that are independent of the key as well as fast. It is also possible to circumvent timing attack by avoiding the use of S-box which will make the AES a bit slow. Another way is the addition of a delay to the faster operations to hide the timing differences but it will also cause performance penalty.

5 Overview of RSA Cryptosystem

In RSA, the plaintext and the cipher text are considered as integers between 0 and n-1, where n is the modulus. The typical size of n is 1024 bits. However, the recommended length of n is 2048 bits as 640 bits key is no more secure by now [2]. The RSA algorithm is comprised of three sub algorithms that are described below:

5.1 Key Generation Algorithm

The key set is generated by using the following algorithm:

1. Select two large prime numbers p and q (e.g. 1024 bits each) such that $p \neq$q.
2. Compute modulus n = p.q
3. Calculate totient, φ(n) = (p-1).(q-1)

4. Choose an integer (public exponent) e, 1 ¡ e ¡ φ(n), such that gcd(e, φ(n)) = 1.

5. Compute the secret exponent d, 1 ¡ d ¡ φ(n), such that $d.e \equiv 1$ (mod n).

The public key is (n, e)
And the private key is (n, d).

5.2 Encryption

Encryption is done by using the following steps:-

1. Obtain the recipient's public key (n, e).

2. Represent the plaintext message as a positive integer m.

3. Compute the cipher text c = m^e mod n.

4. Send the cipher text c to receiver.

5.3 Decryption

Message is decrypted by using the following steps:-

1. Receiver uses his own private key (n, d) to compute m = c^d mod n.

2. Extracts the plaintext from the integer representative m.

6 Computational issues of RSA

1. Selecting the primes p and q:

   In the first step, a random number of —n—/2 bit-length for p is selected. Then the lowest bit and two highest bits are set to ensure p is odd and the highest bit of n will be set. Finally, Miller-Rabin algorithm is applied to ensure that p is a prime number.

   Prime number q is determined following the same steps.

2. Choosing the value of e:

   The mathematical requirement gcd(e, φ(n)) = 1 is equivalent to the

   gcd(e, p-1) = q and

   gcd(e, q-1) = 1 that can be satisfied by choosing a prime number for e. However, for fast modular exponentiation operation, e is chosen among 3, 17 and 65537 as only two bits are set for these three values.

3. Calculating the value of d:

   The requirement $d.e \equiv 1$ (mod φ(n)) is equivalent to d = e^{-1}(mod φ(n)) which is determined by Extended Euclidean Algorithm.

4. Modular Exponentiation for Encryption and Decryption:

   - For encryption: c = m^e mod n can be efficiently computed by selecting an appropriate value for e such as $\{3, 17\ or\ 65537\}$.
   - for decryption: Decryption operation can be efficiently completed by using Chinese Remainder Theorem (CRT) in the following manner:
     c^d mod n = ($v_p x_p$ +$v_q x_q$) mod n
     where , v_p = c^d mod p, v_q = c_d mod q
     and x_p = q (q^{-1} mod p), x_q = p (p^{-1} mod q)
   - Further efficiency can be obtained by applying Fermat's Little Theorem to calculate v_p and v
     v_p = c^d mod p
     = $c^{(u(p-1)+v)}$ mod p
     = $(c^{(p-1)})^u$ c^v mod p
     = $(1)^u$ c^v mod p
     = c^v mod p
     As (v ¡ d), required computation is less than the previous one.

5. Modular exponentiation algorithm:

   Modular exponentiation is the fundamental computation step of RSA that can be effectively performed by using the following formula:

$$A^B mod\ n = \left(\prod b^i \neq 0 \left[\ A^{2^i} mod\ n\ \right]\right) mod\ n$$

7 Security of RSA

The security of the RSA cryptosystem is not perfect. An attacker can resort to different approaches to attack RSA algorithm. Among them, Brute force, Mathematical attacks, Timing attacks and Chosen Ciphertext attacks are covered in brief:

7.1 Brute Force Attack

In a brute force attack, the attacker tries all possible combinations to guess the private key. RSA with short secret key is proven insecure against brute force attack [11].]. This attack can be easily circumvented by choosing large key. However, the larger key will make the encryption and decryption process little slow as it will require greater computations in key generation as well as in encryption/decryption algorithm.

7.2 Mathematical Attacks

Mathematical attack is considered as the most significant type of attack against RSA. By exploiting the mathematical properties of RSA algorithm, an attacker can break the RSA in the following ways:

1. By determining the prime factors p and q of the modulus n, an attacker can findout φ(n) = (p-1)(q-1), which is in turn enables to determine d = e^{-1}(mod φ(n));
2. By figuring out the totient φ(n) directly without first determining p and q from which d = e^{-1}(mod φ(n)) can be determined.
3. By calculating d directly without first determining φ(n).

Mathematical attacks can be prevented by increasing the length of key as security of RSA actually dependant of the difficulty of finding prime factors of it. It is recommended that the size of modulus is 2048 bits.

7.3 Timing Attack

In RSA, the attacker can exploit the timing variation of the modular exponentiation implementations and able to determine d by calculating the time it takes to compute C^d (mod n) for a given cipher text C [13].

There are several countermeasures that can be used against this attack such as:

1. By ensuring constant exponentiation time for all exponentiations. The drawback of this method is it will degrade the computational efficiency of the algorithm.
2. By including a random delay to the exponentiation algorithm that will appear as a noise to the attacker.
3. By multiplying the ciphertext with a random number that will prevent the attacker from bit by bit analysis.

7.4 Chosen Ciphertext Attack

In RSA, the product of two ciphertextsis equal to the encryption of the product of respective plaintexts, i.e. $m_1^e m_2^e$ = $(m_1 m_2)^e$ (mod n). Due to this mathematical property, RSA is vulnerable against the chosen ciphertext attack (CCA) [7].

Example: The ciphertext, c = m^e (mod n) can easily be decrypted by using the following steps

1. Compute x = (c x 2^e) mod n
2. Submit x as a chosen ciphertext and receive y = x^d (mod n)
3. Apply the multiplicative property: x = (c mod n) x (2^e mod n) = (m^e mod n) x (2^e mod n) = $(2m)^e$ mod n

Therefore, y = (2m) mod n from this, the attacker can compute m.

This limitation can be easily solved at the implementation level if the plaintext is padded with the random number. Public Key Cryptography Standards (PKCS) ♯1 defined several versions of RSA Cryptography standard (v1.5, v2.0, v2.1). However, the earlier versions of it (v1-v1.5) have been proven vulnerable to the Adaptive Chosen Ciphertext Attack (CCA2). Later versions of the standard include Optimal Asymmetric Encryption Padding (OAEP), which was introduced by Bellare and Rogway [12], can prevent adaptive chosen cipher text attack.

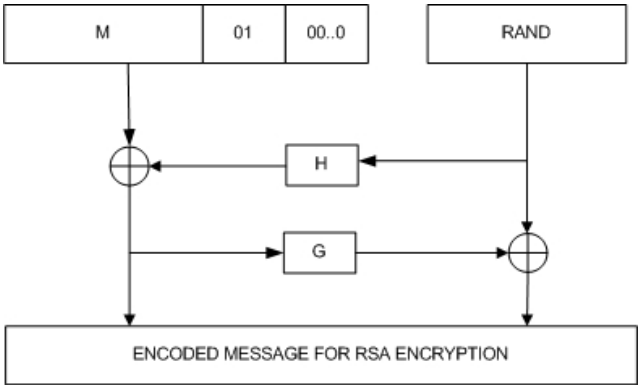

Figure 2. OAEP padding procedure

The OAEP algorithm is a form of Feistel network which uses a pair of random oracles G and H to process the plaintext prior to encryption. OAEP satisfies the following two goals:

1. The deterministic encryption scheme of RSA is replaced by a probabilistic scheme due to the addition of random number.
2. The partial decryption of the ciphertext is prevented unless the attacker is able to invert the trapdoor one-way permutation.

8 Discussion

Some of the conclusions that can be deduced based on the above discussion are specified below:

1. AES is faster and safer encryption algorithm so far compared to other symmetric cryptography. However, it is difficult to write constant-time high-speed AES software for general purpose computers to prevent timing attack. Nevertheless, it is easy to write

slow constant-time software that is immune to this kind of attacks.

2. RSA is also considered safe with large key i.e. 2048 bits. Like AES, implementation may introduce some vulnerability especially when the big number implementation is bad, the big number arithmetic operations in Chinese Remainder Theorem (CRT) may introduce a bug that can be exploited by an attacker to determine the prime factors. This can be circumvented by avoiding CRT which may lead to a slower RSA algorithm.

3. Since RSA (asymmetric cryptography) has more functionality and (AES) symmetric encryption is much faster, it is better to combine these two: Asymmetric cryptography can be used to authenticate the parties and to agree on a key for a symmetric cryptosystem. And then the agreed key will be used for the duration of a single session. This has a number of advantages:

   - Large data blocks are encrypted by the faster AES algorithm rather than the slower RSA and then safely distributed using RSA algorithm.
   - The probability of compromising of these keys is reduced since the amount of encryption with long term keys is minimized.
   - If this key is compromised somehow, the damage will be limited as the session key which is used for encrypting the bulk of the data,is used for only once.

9 Conclusion

In this paper, we have studied AES and RSA encryption schemes and have highlighted some of the important mathematical properties as well as the security issues of both algorithms. Since AES provides better security and has less implementation complexity, it has emerged as one of the strongest and most efficient algorithms in existence today. However, the secret key distribution is considered as a critical issue of AES like other symmetric encryption algorithm. As an asymmetric cryptosystem, RSA solves the problem inherent distributing the secret key. The major drawback of RSA is its greater computational overhead due to its large key. Hence, the optimal solution is the use of a hybrid encryption system in which typically AES is used to encrypt large data block and RSA is used for the key management and digital signature applications.

References

[1] FIPS PUB 197: the official AES standard. http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[2] RSA Data Security, Inc. The RSA Factoring Challenge. http://www.rsa.com/rsalabs/node.asp?id=2092.

[3] A. J. Elbirt,W. Yip, B. Chetwynd, C. Paar. An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9:545–557, 2001.

[4] Adi Shamir Ronald Rivest and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[5] D. J. Bernstein. Cache-timing attacks on AES. April 2005. http://cr.yp.to/antiforgery/cachetiming-20050414.pdf.

[6] J. Bonneau and I. Mironov. Cache-Collision Timing Attacks Against AES. 2005. http://research.microsoft.com/users/mironov/papers/aes-timing.pdf.

[7] D. Bleichenbacher. *A Chosen Ciphertext Attack against Protocols Based on the RSA Encryption Standard PKCS 1; In Crypto 98, LNCS 1462*. Springer-Verlag, 1998.

[8] V. R. Joan Daemen. AES Proposal: Rijndael, version 2, AES submission. 1999. http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf.

[9] Joan Daemen, Steve Borg and Vincent Rijmen. *The Design of Rijndael: AES The Advanced Encryption Standard*. Springer, 2002.

[10] Kofahi, N.A. Turki Al-Somani Khalid Al-Zamil. Performance evaluation of three encryption/decryption algorithms. *Circuits and Systems, 2003. MWSCAS '03. Proceedings of the 46th IEEE International Midwest Symposium on*, 2:790–793, 27-30 December 2003.

[11] M. Wiener. Cryptanalysis of short rsa secret exponents. *IEEE Transactions on Information Theory*, 160:553–558, March 1990.

[12] P. Paillier and J. Villar. Trading one-wayness against chosen-ciphertext security in factoring-based encryption. *Advances in Cryptology - Asiacrypt*, pages 553–558, 2006.

[13] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. *Advances in Cryptology-CRYPTO*, pages 104–113, 1996.

[14] W. D. Smith. 1. AES seems weak. 2. Linear time secure cryptography. *http://eprint.iacr.org/2007/248.pdf*, 2007.

[15] William Stallings. *Cryptography and Network Security Principles and Practices*. Prentice Hall, November 16, 2005.