International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 8, Issue 12, December 2019)

# Comparison of Public Key Cryptography in Different Security Level

Zarni Sann[1], Thi Thi Soe[2], Khaing Myat Nwe[3]
[1,3]*Faculty of Computer Systems and Technologies, University of Computer Studies (Mandalay), Myanmar*
[2]*Faculty of Computer Science, University of Computer Studies (Mandalay), Myanmar*

***Abstract*— Information security is one of the key challenges in data communication. For secure information communication over public network, different cryptographic methods are applied. There have many public key cryptography systems and they have difference performance. This system proposes the performance of the three public key cryptography systems: RSA (Rivest Shamir Adelman), ElGamal and ECC (Elliptic Curve Cryptography) by comparing encryption and decryption messages. This paper presents the implementation and comparison of RSA, Elgamal and ECC for variable text files sizes. Our goal is to calculate encryption time, decryption time, and key size based on different file size for each algorithm to identify which algorithms outperforms others in term of evaluation parameters.**

***Keywords*— Key generation, Encryption, Decryption, RSA, ElGamal, ECC**

## I. Introduction

A public key cryptosystem is an asymmetric cryptosystem where the key is constructed of a public key and a private key. The public key, known to all, can be used to encrypt messages. Only a person that has the corresponding private key can decrypt the message [1].

One of the most widely used public key cryptosystem is RSA. In RSA a public key is constructed by multiplication of two very large primes. To completely break RSA one needs to find the prime factors. In practice, RSA has proved to be quite slow, especially the key generation algorithm. RSA also requires longer keys in order to be secure compared to some other cryptosystems like elliptic curve cryptosystems (ECC) [5].

ECC is one public key cryptosystem, which has proven to be faster than RSA. It is based on a discrete logarithm problem in elliptic curve groups. ECC has been around for some time and has some existing implementations [7].

El-Gamal encryption/decryption is based on the difficulty of the discrete algorithm problem where it is straight forward to raise numbers of large powers but it is much harder to do the inverse computation of the discrete logarithm.

The El-Gamal algorithm depends on certain parameters which are affecting the performance, speed and security of the algorithm. This paper gives an overview of these three cryptosystems and a performance comparison of these cryptosystems based on execution time in different key size and file size [6].

This paper is organized into five sections including this section. Section 2 gives Overview of the cryptography systems. Section 3 describes the background theory of the three public key cryptosystems: RSA, ElGamal and ECC. In section 4 gives the results of the comparison of three cryptosystems with the figure and chats. Section 5 describes the conclusion and further extension of this system.

## II. Cryptography

Cryptography, over the ages, has been an art practiced by many who have devised ad hoc techniques to meet some of the information security requirements. The last twenty years have been a period of transition as the discipline moved from an art to a science.[8] It can be characterized by two types of cryptography systems: (1) Symmetric or private key cryptography system and (2) Asymmetric or public key cryptography system.

### *A. Public Key Cryptography System*

Public-key cryptosystems help solve the key distribution problem by using separate keys for encryption and decryption, and making the encryption key public. Anyone can then encrypt a message, but only parties in possession of the private key can decrypt messages. Public key systems are also known as asymmetric key cryptography [1].

## III. Implementation Methods In The System

In this section, the three public key cryptography systems: Rivest Shamir Adelman (RSA), ElGamal and Elliptic Curve Cryptography (ECC) are presented as implementation of the system.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 8, Issue 12, December 2019)

*B. RSA Encryption*

RSA is widely used in encrypted connection, digital certificates core algorithms. Public key algorithm invented in 1977 by Ron Rivest, Adi Shamir and Leonard Adelman (RSA). It is the main operation of RSA to compute modular exponentiation. Since RSA is based on arithmetic modulo large numbers, it can be slow in constraining environments. Especially, when RSA decrypts the cipher text and generates the signatures, more computation capacity and time will be required. Reducing modules in modular exponentiation is a technique to speed up the RSA decryption. The security of RSA comes from integer to find. Generation of random prime numbers gives the algorithm extra strength and efficiency [2, 5].

*RSA Key Generation*

A RSA public and private key pair can be generated using the algorithm below:

1. Choose two random prime numbers p and q such that the bit length of p is approximately equal to the bit length of q.

$$\text{COMPUTE N SUCH THAT } N = P * Q. \qquad (1)$$

2. Compute f (n) such that

$$\Phi(N) = (P-1)*(Q-1). \qquad (2)$$

3. Choose a random integer e such that $e < \varphi(n)$ and gcd $(e, \varphi(n)) = 1$, then compute the integer d such that:

$$E*D = 1 \text{ MOD } \Phi(N). \qquad (3)$$

4. (n, e) is the public key, and d is the private key.

*RSA Encryption*

Suppose the sender B wishes to send a message (say 'm') to the user A. To encrypt the message using the RSA encryption scheme, the sender B must obtain A's public key pair (e, n). The message to send must now be encrypted using this pair (e, n). However, the message 'm' must be represented as an integer in the interval [0, n-1]. To encrypt it, the user B simply computes the number 'c' where

$$c = m^e \bmod n \qquad (4)$$

B sends the cipher text c to A.

*RSA Decryption*

The receiver A can recover m from c by using A' private key exponent d by the following computation:

$$m = c^d \bmod n \qquad (5)$$

Given m, the receiver A can recover the original message m by reversing the padding scheme.

*C. Elliptic Curve Cryptography (ECC)*

The idea of using Elliptic curves in cryptography was introduced by Victor Miller and N. Koblitz as an alternative to established public-key systems such as DSA and RSA. An elliptic curve E(F p) over a finite field F p is defined by the parameters a, b ∈ Fp. An elliptic curve is given by an equation in the form of:

$$y^2 = x^3 + ax + b \text{ where } 4a^3 + 27b^2 \neq 0. \qquad (6)$$

The set of points on E (F p) also include point O, which is the point at infinity and which is the identity element under addition. This problem is defined as:
Given points X, Y on the elliptic curve, find z such that:

$$X = zY \qquad (7)$$

The elliptic curve parameters for cryptographic schemes should be carefully chosen in order to resist all known attacks of Elliptic Curve Discrete Logarithmic Problem (ECDLP) [2, 3, 7]. Hence, the method of encryption proposed here provides sufficient security against cryptanalysis at relatively low computational overhead.

*ECC Key Generation*

To generate a public and private key pair for use in ECC communications, an entity would perform the following steps:

1. Find an elliptic curve E (K), where K is a finite field such as F p or F2 n, and a find point Q on E (K). n is the order of Q. Recommended domain parameters for E(K) are suggested in [6].
2. Select a pseudo random number x such that $1 \leq x \leq (n-1)$.
3. Compute point $P = x\,Q$. (8)
4. ECC key pair is (P, x), where P is public key, and x is private key.

*ECC Encryption*

To encrypt and send a message $P_m$ to B, the receiver A chooses a random positive integer x and produces the cipher text $C_m$ consisting to the pair of points [5].

$$C_m = \{ xG, P_m + xP_B \} \qquad (9)$$

Note that A has used B's public key $P_B$.

*ECC Decryption*

To decrypt the cipher text, the receiver B multiplies the first point in the pair by B's secret key and subtracts the result from the second point:

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 8, Issue 12, December 2019)

$$P_m + xP_B - n_B(xG) = P_m + x(n_BG) - n_B(xG) = P_m \quad (10)$$

A has masked the message $P_m$ by adding $xP_B$ to it. Nobody but A knows the value of x, so even though $P_B$ is a public key, nobody can remove the mask $xP_B$.

*D. ElGamal Cryptosystem*

The El-Gamal algorithm is a public-key cryptosystem based on the discrete logarithm problem. It consists of both the encryption and signature algorithms. ElGamal algorithm is performed in three parts:

- Key generation for public keys and private keys
- Encryption for original plaintext message to receive cipher text, and
- Decryption for cipher text to generate original plaintext

ElGamal algorithm is illustrated in Figure 2. The key length of the ElGamal can range from 256-bit to arbitrarily long. A key length ranging from 1024 to 2048 bits are considered safe for the next 20 years. Private key can range from 160 bit to 240 bit. ElGamal algorithm is an asymmetric key cryptography, so, it converts message strings to digit using "String to digit conversion table".

*Key Generation*

Key Generation is the first stage of Elgamal algorithm. ElGamal describes the working steps of key generator as follows: [6, 8].

*(1) Select a large prime p;*
*(2) Select d to be a member of the group; (1<=d<=p-2)*
*(3) Select e1 to be a primitive root;*
*(4) e2 = $e1^d$ mod p;*
*(5) public key = (e1, e2, p);*
*(6) private key = d;*
*(7) return public key and private key*

*Elgamal Encryption*

Encryption is the second stage of Elgamal algorithm. ElGamal proves and illustrates the encryption theory in the following algorithm: Message sender chooses a random integer 'r' for encryption and calculates ciphertexts using prime number 'p', random integer 'r' and plaintext 'P', and generates pair of ciphertext for each block (C1,C2). The following algorithm lines are the encryption stage in ElGamal [6, 8].

*(1) Incoming parameter is (e1, e2, p, P)*
*(2) Select a random integer r*
*(3) C1 = $e1^r$ mod p*
*(4) C2 = (P x $e2^r$) mod p*
*(5) return (C1, C2)*
*(6) (C1 and C2 are ciphertexts)*

Note that one can easily find $h^y$ if one knows m'. Therefore, a new y is generated for every message to improve security. For this reason, y is also called an ephemeral key.

*Elgamal Decryption*

Decryption is the third stage of Elgamal algorithm. ElGamal proves and illustrates the decryption theory in the following algorithm. The recipient may retrieve the message by using public keys and private keys. Decryption process uses ciphertexts to generate plaintext, again. The result of decryption process from the ElGamal algorithm and original incoming messages are the same. The following algorithm lines are the decryption stage in ElGamal [6, 8].

*(1) Incoming parameter is (d, p, C1, C2).*
*(2) P = [C2 $(C1^d)^{-1}$] mod p*
*(3) return P*
*(4) (P is the plaintext).*

The result of decryption process from the ElGamal algorithm and original incoming messages are the same.

## IV. Experimental Results Of RSA, Elgamal And ECC

Authors can select following parameters for evaluation of RSA, ElGamal & ECC asymmetric encryption algorithms for both encryption and decryption schemes. In this system, to test and compare the performance characteristics of the RSA, ElGamal and ECC algorithms, independently tested each of the four main components:, encryption time and file encryption, decryption time and file decryption on different key size.

(1) *Encryption time (Computation Time/Response Time):* The encryption time is considered the time that an encryption algorithm takes to produces a cipher text from a plain text.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 8, Issue 12, December 2019)

(2) *Decryption time (Computation Time/Response Time):* The decryption time is considered the time that an encryption algorithm takes to reproduces a plain text from a cipher text.
(3) *Encrypted File Size:* The size of encrypted file is called encrypted file size.
(4) *Decrypted File Size:* The size of decrypted file is called decrypted file size [4].

Since ECC offers security equivalent to RSA and ElGamal using much smaller key sizes, the performances were tested according to the following table 1.

**TABLE I**
**COMPARABLE KEY SIZES (IN BITS)**

| Security Level | RSA/ ElGamal | ECC |
|---|---|---|
| Low | 512 | 112 |
| Medium | 1024 | 160 |
| High | 2048 | 224 |

The test results show the key generation time of ECC is faster than the key generation time of RSA and Elgamal.

### *E. Encryption Time Comparison of RSA, Elgamal and ECC*

Table 2 is the table of encryption time at low security level. The test results show that the encryption time of RSA and ECC is not much different when the encrypt file is small. But the file size increases, the encryption time is more difference. It shows that when large file are encrypted, ECC gains increasing advantage in performance. Table 3 shows the encryption time at medium security level.

**TABLE II**
**ENCRYPTION TIME OF RSA AND ECC AT LOW LEVEL SECURITY**

| File size (KB) | Low Level Encryption Time (ms) | | |
|---|---|---|---|
| | RSA | Elgaml | ECC |
| 25 | 1656.25 | 1856.2 | 1358.17 |
| 50 | 2065.62 | 2205.6 | 1436.21 |
| 75 | 2458.73 | 2656.1 | 1514.61 |
| 100 | 2984.37 | 3101.2 | 1600.97 |
| 500 | 4917.86 | 5102.8 | 1946.40 |

**TABLE III**
**ENCRYPTION TIME OF RSA AND ECC AT MEDIUM LEVEL SECURITY**

| File size (in KB) | Medium Level Encryption Time (ms) | | |
|---|---|---|---|
| | RSA | Elgamal | ECC |
| 25 | 1796.87 | 1850.6 | 1467.74 |
| 50 | 2203.75 | 2346.2 | 1545.51 |
| 75 | 2598.18 | 2720.6 | 1623.30 |
| 100 | 3121.06 | 3205.6 | 1710.61 |
| 500 | 5432.69 | 5896.6 | 2177.34 |

The encryption time for ECC is shown to be faster than RSA because the size of the encrypted data depends on the size of the key and the size of data. In ECC, the size of the encrypted data depends on the key size and the input data size. Table 4 shows the encryption time at high security level.

**TABLE IV**
**ENCRYPTION TIME OF RSA AND ECC AT HIGH LEVEL**

| File size (KB) | High Level Encryption Time (ms) | | |
|---|---|---|---|
| | RSA | Elgamal | ECC |
| 25 | 1944.37 | 1858.6 | 1571.87 |
| 50 | 2231.25 | 2306.2 | 1655.75 |
| 75 | 2731.62 | 2739.6 | 1737.65 |
| 100 | 3264.57 | 3235.8 | 1818.25 |
| 500 | 5434.54 | 5906.2 | 3464.34 |

### *F. Decryption Time Comparison of RSA and ECC*

Table 5 shows the decryption time of RSA, Elgamal and ECC at low security level. Table 6 shows the encryption time at medium security level. Table 7 shows the decryption time at high security level. The test results show the decryption of ECC is faster than RSA. It can also be seen that as the file size increases, ECC gains increasing advantage in performance.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 8, Issue 12, December 2019)

**TABLE V**
**DECRYPTION TIME OF RSA AND ECC AT LOW LEVEL SECURITY**

| File size (KB) | Low Level Decryption Time (ms) | | |
|---|---|---|---|
| | RSA | Elgamal | ECC |
| 25 | 2516.52 | 2601.5 | 1094.71 |
| 50 | 4117.67 | 3850.2 | 1139.1 |
| 75 | 5719.34 | 4855.4 | 1174.49 |
| 100 | 7381.38 | 7201.6 | 1215.78 |
| 500 | 13714.89 | 13779.5 | 1546.65 |

**TABLE VI**
**DECRYPTION TIME OF RSA AND ECC AT MEDIUM LEVEL SECURITY**

| File size (KB) | Medium Level Decryption Time (ms) | | |
|---|---|---|---|
| | RSA | Elgamal | ECC |
| 25 | 2859.37 | 2701.5 | 1467.76 |
| 50 | 4460.29 | 4106.2 | 1508.29 |
| 75 | 6062.86 | 6120.6 | 1551.14 |
| 100 | 7726.68 | 7433.0 | 1591.83 |
| 500 | 8406.74 | 11503.4 | 1755.68 |

**TABLE VII**
**DECRYPTION TIME OF RSA AND ECC AT HIGH LEVEL SECURITY**

| File size (KB) | High Level Decryption Time (ms) | | |
|---|---|---|---|
| | RSA | Elgamal | ECC |
| 25 | 3208.37 | 2820.5 | 1571.83 |
| 50 | 4803.25 | 4210.6 | 1884.74 |
| 75 | 6405.94 | 6312.2 | 1925.49 |
| 100 | 8071.75 | 7623.1 | 1965.51 |
| 500 | 14321.89 | 16521.0 | 2243.57 |

## V. CONCLUSION

Public key cryptography has become an important means of ensuring confidentiality, notably through its use of key distribution. Key distribution is an approach where users seeking private communication exchange encryption keys, while digital signatures allow users to sign keys to verify their identities. The ElGamal algorithm can be used as RSA algorithm for public key encryption because:

RSA encryption depends on the difficulty of factoring large integers while ElGamal encryption depends on the difficulty of computing discrete logs in a large prime modulus. By comparing the performance of ECC with RSA in terms of key generation time, encryption time and decryption time for the same level of security, ECC is faster than key generation time, encryption time and especially in decryption time. The smaller key sizes of ECC potentially allow for less computationally able devices such as smart cards and embedded systems to use cryptography for secure data transmissions, message verification and other means. ElGamal ‘s encryption is very simple because it is multiplication of message and symmetric key(i.e c=m*k).

This study could be extended by comparing other Public Key Cryptography systems. This system was focused on time of the operations, other factors should be considered such as memory space, CPU utilization, different key size and also different type of attack protection for security level.

## REFERENCES

[1] Anita Ganpati, Narender Tyagi, A Survey of Different Public-Key Cryptosystems, International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 6, Nov-Dec 2015

[2] B.Arrendondo and N. Jansma, “Performance Comparison of Elliptic Curve and RSA Digital Signatures”, April 28, 2004.

[3] I. Blake, G. Seroussi and N. Smart, Elliptic Curves in Cryptograph, Cambridge University Press, 1999

[4] J. Loikkanen and P. Karu, “Practical Comparison of Fast Public-key Cryptosystems”, Helsinki University of Technology, 2000.

[5] P. Riikonen,“RSA Algorithm”, April 28, 2004. http://iki.fi/priikone/docs/rsa.pdf

[6] Shaina Arora, Pooja, Enhancing Cryptographic Security using Novel Approach based on Enhanced-RSA and Elamal: Analysis and Comparison, International Journal of Computer Applications (0975 – 8887) Volume 112 – No 13, February 2015

[7] Sougata Khatua and N.Ch.S.N. Iyengar, Analyzing the Security System Applied in E-Shopping System Using Elliptic Curve Cryptography, International Journal of Latest Trends in Computing (E-ISSN: 2045-5364) Volume 2, Issue 2, June 2011

[8] W. Stallings, Cryptography and Network Security, Prentice Hall, Second Edition, 1998.