Journal of Theoretical and Applied Information Technology
10th January 2013. Vol. 47 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEW COMPUTATION TECHNIQUE FOR ENCRYPTION AND DECRYPTION BASED ON RSA AND ELGAMAL CRYPTOSYSTEMS

ZULKARNAIN MD ALI[1] AND JASSIM MOHAMMED AHMED[2]
School of Computer Science, Faculty of Information Science and Technology,
University Kebangsan Malaysia, 43600 Bangi,
Selangor Darul Ehsan, Malaysia

Email: [1]zma@ftsm.ukm.my, [2]jasimgm76@yahoo.com

Received September 2012

ABSTRACT

Cryptography addresses the necessary elements for secure communication that involved privacy, confidentiality, key exchange, authentication, and non-repudiation. This paper proposed a new computation of encryption and decryption that was based on the computation of RSA and ElGamal Cryptosystems. The RSA is a public key encryption system that gets its security from the difficulty of factoring large numbers. Meanwhile, the security of ElGamal is depends upon the difficulty of a certain problem that was related to compute Discrete Logarithms. The proposed technique shows better computation time compared to both cryptosystems. All computation time for each algorithm also presented at the end of this paper. However, this paper did not present the security of the technique over the RSA and ElGamal.

Keywords: *RSA, ElGamal, Encryption and Decryption*

1. INTRODUCTION

Communication is the activity of conveying meaningful information. The communication process is complete once the receiver has understood the message of the sender. In today's communication, the security issues always been the top priority. Cryptography is the most common necessary elements for secure communication based on privacy, confidentiality, key exchange, authentication, and non-repudiation but reveals the fact that communication is happening [7]. This paper introduces a new computation technique that satisfies the security requirement for sending the confidential message.

The RSA cryptosystem is depend on the Integer Factorization Problem (IFP). The integer factorizations or prime factorization is the decomposition of a composite number into smaller non-trivial divisors, which when multiplied together equals the original integer. The security of RSA is depends on the difficulty of the process of factorization. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message [5][9].

On the other hand, the ElGamal cryptosystem depend on Discrete Logarithm Problem (DLP). The security of the ElGamal scheme depends on the properties of the underlying group G as well as any padding scheme used on the messages [10]. The ElGamal encryption is unconditionally malleable, and therefore is not secure under chosen cipher text attack.

To achieve chosen-ciphertext security, the scheme must be further modified, or an appropriate padding scheme must be used. ElGamal encryption is probabilistic, meaning that a single plaintext can be encrypted to many possible ciphertexts, with the consequence that a general ElGamal encryption produces a 2:1 expansion in size from plaintext into ciphertext [3]. Encryption under ElGamal requires two exponentiations; however, these exponentiations are independent of the message and can be computed ahead of time if need be. Decryption only requires one exponentiation.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This paper manipulates both cryptosystems computation. A big size of keys is needed to give high level of security of these systems [1]. Using a large key size causes the system requires extensive and continuous computation and high processing capabilities. The larger the size the more difficult of encryption and decryption processes [4]. Therefore, this paper aims to use some of the properties in the RSA and ElGamal to produce a better technique in terms of computation efficiency. In order to show the efficiency of the proposed scheme, the comparison of computation time between these three algorithms is included. Furthermore, the mathematical proof needed to show the similarities and differences between the new technique and the existing techniques.

2. REVIEW ON RSA AND ELGAMAL CRYPTOSYSTEMS

The art of hiding messages in unreadable secret codes is called as cryptography. This technique concentrates on preventing messages from being reproduced, stolen and damaged [2]. The cryptosystem is pair of algorithms that take a key and convert plaintext to ciphertext and back. Both RSA and ElGamal cryptosystems are the most popular asymmetric cryptosystems, but it needs a relatively long time in the process of encryption and decryption.

2.1. RSA Cryptosystems

Rivest, Shamir and Adleman have introduced the RSA cryptosystem in 1978. It was first publicly known as a public-key cryptosystem. A detail discussion on this cryptosystem can also be found in [5].

For the past 3 decades, the RSA has become the most common standard of encryption in the computer world. In fact, the RSA is now the most public-key system in wide use. The reason behind this is that the RSA now used in many web browsers such as Netscape to E-mail encryption programs, in another word RSA is now everywhere [6][8].

The RSA algorithm can be described as follows:

i. Key generation:
   a. Choose two distinct prime numbers p and q.
   b. Compute n = pq.
   c. Compute $\varphi(n) = (p-1)(q-1)$, where φ is Euler's totient function.
   d. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e,\varphi(n)) = 1$, i.e. e and $\varphi(n)$ are coprime.
   e. Determine $d = e^{-1} \bmod \varphi(n)$; i.e. d is the multiplicative inverse of e mod $\varphi(n)$.

ii. Encryption:
The sender computes $C=M^e \bmod n$. A C is the encrypted messages.

iii. Decryption:
The receiver uses her private key to compute $M = C^d \bmod n$, where M is the original message [5] .

2.2. ElGamal Cryptosystems.

Meanwhile, the ElGamal Cryptosystem is a very successful implementation of Diffie-Hellman algorithm [9]. Taher ElGamal in 1985 proposed El-Gamel cryptosystem algorithm in a type of public-key cryptosystem algorithm which is used over finite fields and its security is based on the Discrete Logarithm Problem (DLP) [9]. The reason behind using DLP is due to the difficulty of finding discrete logarithm while the inverse operation of exponentiation can be compute in a straight-forward fashion.

The ElGamal public key system uses modular exponentiation as the bases of a one way trap door function [11]. This is making ElGamal an attractive alternative to the more well known RSA system. The ElGamal algorithm can be described as follows:

i. Key Generation:
   a. Choose a random prime $\mathbb{P}$.
   b. Compute a random multiplicative generator element g of f_p^*.
   c. Choose a random number $x \in Uf_{p-1}$ as the private key.
   d. Compute the public key by $y \leftarrow g^x \pmod p$.
   e. Make (p, g, y) public, and keep (x) as private key.

ii. Encryption:
The sender picks $k \in Uf_{p-1}$ at random and computes a cipher text pair (c1,c2): $c_1 \leftarrow g^k \pmod p$ and $c_2 \leftarrow g^x \pmod p$

iii. Decryption:
To decrypt cipher text (c_1, c_2), the receiver computes $m \leftarrow c_2/c_1^x \pmod p$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PROPOSED COMPUTATION TECHNIQUE.

The proposed computation techniques merge the RSA and ElGamal computations. This computation technique makes use of some properties of RSA and ElGamal. As a result, this computation technique provides better and efficient computation time.

3.1 Initialization

i. Alice and Bob publicly choose prime q a finite field Fq. They publicly choose a random base element g ∈ Fq.
ii. The receiver chooses two large prime numbers P and Q. Compute n=P*Q. Publicly choose a random base.
iii. The key generation of the proposed method is as follows:

Step 1: Choose $1<e<\varnothing(n)$, $\varnothing(n)=(P-1)*(Q-1)$, with gcd $(e,(P-1)*(Q-1))=1$.
Step 2: Compute $d=e^{-1} \bmod (P-1)*(Q-1)$
Step 3: Make n, g, q and e public and keep private key d secrets.

3.2 Encryption

i. The senders sends the message M to Bob where the message M as an integer element, M<q.
ii. Steps involved to convert plaintext to cipher text:

Step 1: Select random integer a such that 1<a<q
Step 2: Compute $k= g^a \bmod q$.
Step 3: Compute $C_M=M*k \bmod q$ and also $C_a= a^e \bmod n$
Step 4: Transmit pair of ciphertext (C_M,C_a).

3.3 Decryption

The receiver does the following computations:

Step 1: Compute $a = c_a^d \bmod n$.
Step 2: Compute $k =g^a \bmod q$.
Step 3: Compute $k^{-1} =g^{-a} \bmod q$.
Step 4: Compute $M = C_M * k^{-1} \bmod q$.

Proof:

Initialization:

a. n=P*Q
b. Choose e, where $1<e<\varnothing(n)$, $\varnothing(n)=(P-1)*(Q-1)$, with gcd(e,(P-1)*(Q-1))=1.
c. Compute: $d= e^{-1} \bmod (P-1)*(Q-1)$ where n, g, q and e public and keep private key d secret.

Encryption

a. M < q
b. $k = g^a \pmod q$
c. $C=M*k \bmod n =M*g^a =M*g^a \bmod n$

Decryption

a. $M = C*k^{-1} \bmod n$
b. $k^{-1} = g^{-a} \pmod n$
c. $k^{-1} = g^{-a} \pmod n = C*g^{-a} \pmod n$

Therefore,

$M = C*g^{-a} \bmod n$
$= [M * (g^a)] * (g^{-a}) \bmod n$
$= M * (g^a) *(g^a)^{-1} \bmod n$
$= M.$
■

4. EXAMPLE OF COMPUTATION TECHNIQUES FOR EACH ALGORITHM

4.1 Example For RSA.

i. Initialization:
- Bob choose two large prime numbers: P = 997 and Q=1171.
- N=P*Q

ii. Key generation:
- Bob compute n = P*Q = P*Q = 997*1171 = 1167487.
- Choose e, where $1 < e < \varnothing(n)$, with $\varnothing(n)=(P-1)(Q-1)$ and gcd(e,(P-1)(Q-1))=1
- Then, $\varnothing(n) = (P-1)(Q-1) = 1165320$
- Let e=383, $1 < 383 < 1165320$
- Compute $d = e^{-1} \bmod \varnothing(n) = 383^{-1} \bmod 1165320 = 54767$
- Make n, e public and keep d secret.

iii. Encryption:
- Alice wants to send a message M to Bob.
- She do encryption as follow:
  - Let M = 47964
  - Compute $C = M^e \bmod n = 47964^{383} \bmod 1167487 = 966090$
  - Transmit 966090.

iv. Decryption:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Bob retrieves the message send by Alice by decryption as follows:
  - Compute $M = C^d \bmod n$
  - $M = 966090^{54767} \bmod 1167487 = 47964$

4.2 Example For ElGamal.

i. Initialization:
- Alice and Bob publicly choose prime q a finite field F_q: q = 9369319.
- They publicly choose a random base element $g \in F_q$: g = 1632.
- Choose b, where 1<= b <= q-2 , and compute $y_b = g^a \bmod q$

ii. Key generation:
- Choose b, where 1 <= b <= 9369317
- Let b=51
- Compute $y_b = g^a \bmod p = 1623^{51} \bmod 9369319 = 8121285$
- Make y_b and q public and keep d secret.

iii. Encryption:
- Alice wants to send a message M to Bob.
- She do encryption on message as follows
  - Let M = 47964.
  - Let k=10
  - Compute $y_a = g^k \bmod p = 162310 \bmod 9368319 = 8932195$
  - Compute ciphertext $C = M*y_b^k \bmod q = 47964*8121285^{10} \bmod 9369319 = 5057598$
- Transmit the pair (8932195,5057598)

iv. Decryption:
- Bob retrieves the message send by Alice by decryption as follow:
- Bob actually retrieves (y_a,C) as follows:
  - $y_a^{-b} \bmod q = 8932195^{51} \bmod 9369319 = 2927139$
  - $M = C*y_a^{-b} \bmod q = (c*2927139) \bmod q = (5057598*2023) \bmod 9369319 = 47964$

4.3 Example 1 For Proposed Technique.

i. Initialization:
- Alice and Bob publicly choose prime q a finite field F_q: q = 9369319.
- They publicly choose a random base element $g \in F_q$: g = 1632.
- Bob choose two large prime numbers: P=997 and Q= 1171.

ii. Key generation:
- Compute n = P*Q = 997*1171 = 1167487.
- Choose e, where 1 < e < ∅(n), with, ∅(n) = (P-1)(Q-1) and gcd(e, ∅(n)) = 1.
- ∅(n) = (P-1)*(Q-1) = 1165320.
- Let e=383, because 1 < 383 < 9369319
- Compute $d = e^{-1} \bmod ∅(n) = 383^{-1} \bmod 1165320 = 54767$
- Make n, e, q public and keep d secret.

iii. Encryption:
- Alice wants to send a message M to Bob.
- She do encryption on message as follows:
  - Let M = 47964.
  - Select random integer a = 51
  - Compute $k = g^a \bmod q = 1623^{51} \bmod 9369319 = 8121285$
  - Then compute ciphertext C, C = M*k mod n = 47964*689466 mod 9369319 = 9245634
  - Also compute $a_c = a^e \bmod n = 51^{383} \bmod 1167487 = 356885$
- Transmit the pair points (356885,9245634).

iv. Decryption:
- Bob retrieves the message send by Alice by decryption as follow:
  - Compute $a = a_c^d \bmod n = 356885^{54767} \bmod 1167487 = 51$
  - Compute $k = g^a \bmod q = 1623^{51} \bmod 9369319 = 8121285$
  - Compute $k^{-1} = g^{-a} \bmod q = 689466^{-1} \bmod 9369319 = 1111728$
  - Compute M = C*k-1 mod n = 477949*1111728 mod 9369319 = 47964

4.4 Example 2 For Proposed Algorithm.

i. Initialization:
- Alice and Bob publicly choose prime q a finite field F_q: q = 9369319.
- They publicly choose a random base element $g \in F_q$: g = 1632.
- Bob choose two large prime numbers: P=997 and Q= 1171.

ii. Key generation:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Compute n = P*Q = 997*1171 = 1167487.
- Choose e, where 1 < e < ∅(n), with, ∅(n) = (P-1)(Q-1) and gcd(e, ∅(n)) = 1.
- ∅(n) = (P-1)*(Q-1) = 1165320.
- Let e=383, because 1 < 383 < 9369319
- Compute d = e^{-1} mod ∅(n) = 383^{-1} mod 1165320 = 54767
- Make n, e, q public and keep d secret.

iii. Encryption:

- Alice wants to send a message M to Bob.
- She do encryption on message as follows:
  - Let M = 1976.
  - Select random integer a = 97
  - Compute k = g^a mod q = 1623^{97} mod 9369319 = 4282719
  - Then compute ciphertext C, C = M*k mod n = 1976*4282719 mod 9369319 = 2157687
  - Also compute a_c = a^e mod n = 97^{383} mod 1167487 = 618993
- Transmit the pair points (618993,2157687).

iv. Decryption:

- Bob retrieves the message send by Alice by decryption as follow:
  - Compute a = a_c^d mod n = 618993^{54767} mod 1167487 = 97
  - Compute k = g^a mod q = 1623^{97} mod 9369319 = 4282719
  - Compute k^{-1} = g^{-a} mod q = 4282719 mod 9369319 = 7313836
  - Compute M = C*k-1 mod n = 2157687*7317836 mod 9369319 = 1976

5. RESULTS AND DISCUSSION.

5.1. Computational Complexity.

a. RSA

The encryption scheme of RSA is C=M^e mod n

Therefore, the complexity of encryption process takes: T(C)=O(log n) ^{3} bit operation.

Meanwhile, the decryption scheme of RSA is M=C^d mod n
Therefore, the complexity of decryption process takes: T(M)=O(log n) ^{3} bit operation.

b. ElGamal

Let the size of the input message unit is n.
The encryption scheme of ElGamal is k=q^a mod n and h=g^a mod n, (that is sending to the receiver to compute k)

C=M*k .

T(k)=O(log n) multiplication operation, using fast exponential method, where each multiplication operation take O(log n) ^{2}.

Computation of T(k)=O(log n) O(log n)^2 bit operation. It is also applied to T(k)= O(log n)^3 bit operation. T(h)= O(log n)^3 bit operation. The multiplication of M*k take O(log n).

Therefore the complexity of ElGamal encryption process takes: T(C)=O(log n) + 2 O(log n)^3 bit operation.

Meanwhile, the decryption scheme of ElGamal is:
k=h^b mod n

Compute: k^{-1} mod n,
M=C * k^{-1} mod n.

T(k)=O(log n) multiplication operation, using fast exponential metod, where each multiplication operation take O(log n)^2. T(k)=O(log n) O(log n)^2 bit operation. T(k)= O(log n)^3 bit operation. The computation of k^{-1} take O(log n)^3, by extend Euclid's method. T(k^{-1}) = O(log n)^3, the multiplication of C*k^{-1} take O(log n).

Therefore, the complexity of ElGamal decryption process takes: T(C)= O(log n) + 2O(log n)^3 bit operation.

iv. The Proposed Technique.

The encryption scheme of the proposed method is:
k = g^a mod n
a_c = a^e mod n, (that is send to the reciever to compute k)

C=M*k .

Then: T(k)= O(log n)^3 bit operation. T(a_c)= O(log n)^3 bit operation.

Therefore, the complexity of encryption process takes: T(C)= O(log n) + 2 O(log n)^3 bit operation.

Meanwhile, the decryption scheme of the proposed method is:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$a = a_c^d \bmod n$
$k = g^a \bmod n$
$k^{-1} \bmod n$,
$M = C*k^{-1} \bmod n$.

Then: $T(a_c)= O(\log n)^3$ bit operation. $T(k)= O(\log n)^3$ bit operation. $T(k^{-1})= O(\log n)^3$ bit operation.
Therefore, the complexity of decryption process takes: $T(M)= O(\log n) + 3O(\log n)^3$ bit operation.

Table 1: Computational Complexity For Each Algorithm

Algorithm	Computational Complexity
RSA	$O(\log n)^3$
ElGamal	$O(\log n)+ 2O(\log n)^3$
Proposed	$O(\log n) + 3\ O(\log n)^3$

5.2. Computational Time.

All these algorithms have been implemented in MATLAB 7. The completed programs have been tested on PC computer with Windows Vista™ Home Premium. The PC has Intel® Core™ 2 Duo CPU T5750 @ 2000GHz 2000GHz. It also has RAM up to 3.00GB. The text data has been tested in various sizes.

The computation time for encryption and decryption on all algorithms are shown in Table 2 below. It is clearly shown that, the proposed algorithm shows slightly better computation time compare to RSA and ElGamal.

Message KB)Sizes	Operation	GamalEl (sec)	RSA (sec)	Proposed (sec)
1	Encrypt	0.12	0.15	**0.120**
	Decrypt	0.12	0.12	**0.120**
2	Encrypt	0.25	0.292	**0.240**
	Decrypt	0.24	0.262	**0.240**
4	Encrypt	0.51	0.62	**0.503**
	Decrypt	0.50	0.57	**0.494**
10	Encrypt	1.45	1.78	**1.454**
	Decrypt	1.44	1.67	**1.454**
20	Encrypt	3.52	4.35	**3.511**
	Decrypt	3.50	4.25	**3.503**
40	Encrypt	11.81	17.32	**11.74**
	Decrypt	11.71	22.0	**11.50**
100	Encrypt	169	195	**166**
	Decrypt	168	250	**165**

6. CONCLUSION

The proposed method is a combination between RSA and ElGamal cryptosystem based on its problems. These algorithms give additional difficulties, because the employment of DLP and IFP in the proposed algorithm. The complexity and the execution time of the public-key cryptosystems are trade-off problem. The execution time of the proposed method doesn't differ great than the original methods. As the RSA algorithm depends on the (DLP) and algorithm EIGamel depends on (IFB), the proposed system resulting from the merger between these algorithms is becoming more complicated than each algorithm operates to work independently. As a consequence, the preservation is done on the time of encryption and decryption gives the system more sophistication, and strength without any loss of time compared to that of RSA or ElGamal.

ACKNOWLEDGEMENTS

The authors would like to express gratitude to Kementerian Pengajian Tinggi Malaysia and Universiti Kebangsaan Malaysia for a research grant FRGS/1/2012/SG05/UKM/02/1.

REFERENCES

[1] A. E. Cohen, "Architectures for Cryptography Accelerators," UNIVERSITY OF MINNESOTA, 2007.
[2] A. K. Ho, "Hybrid Cryptosystem Using Symmetric Algorithms and Public-key Algorithms," Lamar University., 1999.
[3] B. Allen, "Implementing several attacks on plain ElGamal encryption," Ames, Iowa, Iowa 2008.
[4] Paar, C. 2000. Applied cryptography and data security. Lecture Notes), Ruhr-Universität Bochum (http://www. crypto. ruhr-uni-bochum. de).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[5] Rivest, R.; A. Shamir; L. Adleman (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the ACM 21 (2): pp120–126.

[6] S. Galbraith, et al., "Tunable balancing of RSA. Information security and privacy," 2005, pp. 280–292.

[7] S. Inshi and A. Youssef, "Design and implementation of an online anonymous feedback system," in Communications, 2008 24th Biennial Symposium on, 2008, pp. 58-61.

[8] S. Robinson, "Still guarding secrets after years of attacks, rsa earns accolades for its founders," SIAM News, vol. 36, pp. 1-4, 2003.

[9] S. Upadhyay (2011), "Attack on RSA Cryptosystem", International Journal of Scientific & Engineering Research (IJSER), Volume 2, Issue 9, September 2011. pp 1-4.

[10] Taher ElGamal (1985). "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". IEEE Transactions on Information Theory 31 (4): pp 469–472.

[11] W. Mao, Modern cryptography: theory and practice: Prentice Hall Professional Technical Reference, 2003.