International Journal of Computer Applications Technology and Research
Volume 7–Issue 02, 40-44, 2018, ISSN:-2319–8656

An Improvement of the Basic El-Gamal Public Key Cryptosystem

W.D.M.G.M. Dissanayake
(PG/MPhil/2015/09)
Department of Computer Engineering
Faculty of Engineering, University of Peradeniya, Sri Lanaka

Abstract: In this paper an improvement of the El-Gamal public key cryptosystem is presented. The public key of the El-Gamal system is not changed in this method. But, the sending structure of message and the decryption process are changed. The El-Gamal cryptosystem is not secure under adaptive chosen ciphertext attack. That means El-Gamal cryptosystem can be ciphertext attacked without knowing any key. Therefore changing keys of El-Gamal cryptosystem are not useful. This improvement cryptosystem immunes against CPA and CCA attacks. This cryptosystem is practical and very simple. The importance of this modified cryptosystem is any adversary can't find the sending message in easily.
Keywords: public key cryptosystem, RSA public key cryptosystem, El-Gamal public key cryptosystem, Elliptic Curves Cryptosystem, chosen ciphertext attack, chosen plaintext attack

1 INTRODUCTION

Since the public key cryptography was introduced by Diffie and Hellman in 1976, designing Public Key Crypto Systems is very important research area in world. RSA cryptosystem, El-Gamal cryptosystem and Elliptic Curves cryptosystem are famous public key cryptosystems. But, there is no guarantee for the security of any cryptosystem yet. For an example anyone can attack to the ciphertext of El-Gamal cryptosystem without knowing any keys. Many countries are trying to find a better cryptosystem and fund more to research projects based on cryptography. There are many public key cryptosystems have been developed in world. But, we can't trust 100% none of those systems.

I describe here briefly the definition of public key cryptosystem and two famous public key cryptosystems in world, the RSA public key cryptosystem and the El-Gamal public key cryptosystem.

1.1 Definition:

A public key cryptosystem is a tuple of probabilistic polynomial –time algorithm $(Kgen, Enc, Dec)$ such that:

1. $Kgen$ is a probabilistic key generation algorithm that takes as input 1^k for a security parameter $k \in \mathbb{N}$ and returns a public key pk and a secret key sk. The public key *pk* defines a space *M*, called message space.
2. *Enc* is a probabilistic algorithm that takes as input a public key pk and a message $m \in M$ and returns a ciphertext *c*.
3. *Dec* is a deterministic algorithm that takes as input a secret key sk and a ciphertext *c*, and returns a message m or the reject symbol $\perp$. Moreover a further fundamental property is required: correctness. We want that for every $k \in \mathbb{N}$, every pair $(pk, sk) \leftarrow Kgen(1^k)$, and for every message $m \in M$, the following equation holds:
   $\Pr[Dec(sk, Enc(pk, m)) = m] = 1$.

1.2 RSA public key cryptosystem

This public key cryptosystem was introduced by R.L. Rivest, A. Shamir and L. Adleman in 1978. This system was the first practical public key cryptosystem. Following is the RSA scheme.

1. Two large prime numbers are generated. Let *p* and *q*.
2. Modulus *n* is generated by multiplying *p* and *q*.
3. The totient of *n* is $\phi(n) = (p-1).(q-1)$ is calculated.
4. Public Key: A prime number *e* is selected. where $3 \leq e \leq \phi(n)$ and gcd $[e, \phi(n)] = 1$; gcd means greatest common divisor.
5. Private Key: The inverse of *e* with respect to mod $\phi(n)$ is calculated.

   The RSA function for message *m* and key *k* is,

   $F(m, k) \equiv m^k \bmod n$

   Encryption: $m^e \bmod n \equiv c$

   Decryption: $c^d \bmod n \equiv m$

Example: Let $p = 7$ and $q = 11$.
Then $n = 77$ and $\phi(n) = 60$
Choose $e = 13$. gcd $[e, \phi(n)] = 1$,
Then the secret key *d* can find easily. $e.d \equiv 1 \bmod \phi(n)$
i.e. $13.37 = 1 \bmod 60$, Hence, $d = 37$.
Let the message is $m = 6$
Encryption: $m^e \bmod n \equiv 6^{13} \bmod 77 \equiv 62 \equiv c$
Decryption: $c^d \bmod n \equiv 62^{37} \bmod 77 \equiv 6 \equiv m$
The security of RSA is based on the infeasibility of factorization large *n*.

1.3 The El-Gamal cryptosystem

This public key cryptosystem was introduced by Taher Elgamal in 1985.

Step 01: Global elements: Let any large prime number *p* and a primitive root g of *p*.

Step 02: Decryption key: x – private, Calculate $g^x \bmod p$, where $x \in \mathbb{Z}$.

Publish $(p, g, g^x \bmod p)$.

Step 03: Encryption:

Let the message is *m*; $(0 < m < p)$ and choose *y* - private $(0 < y < p)$.

Compute $b = g^y \bmod p$. Then,

$c \equiv m.a^y \bmod p$.

Send (*b*, *c*).

Step 04: Decryption:

Compute $b^x \bmod p \equiv a^y$. Then,

$m \equiv a^{y^{-1}}.c \bmod p$.

Example:

Step 01: Select $p = 23$ and a primitive root of $p = 23$ is $g = 5$.

Step 02: Let, $x = 8$.

Calculate $g^x \bmod p \equiv 5^8 \bmod 23 \equiv 16$.

Publish $(23, 5, 16)$.

Step 03: Encryption:

Let the message is $m = 6$; and choose $y = 3$

Compute $b \equiv g^y \bmod p \equiv 5^3 \bmod 23 \equiv 10$.
Then,
$c \equiv m.a^y \bmod p \equiv 6.16^3 \bmod 23 \equiv 12.$
Send (*10, 12*).

Step 04: Decryption:
Compute $b^x \bmod p \equiv 10^8 \bmod 23 \equiv 2.$
Then,
$2^{-1}.12 \bmod 23 \equiv 6 \equiv m$.

The security of El-Gamal cryptosystem is depended on the discrete logarithm problem.

1.4 A chosen ciphertext attack on El-Gamal public key cryptosystem

The El-Gamal system is not secure under Chosen Ciphertext Attack. Anyone can easily get the message.

Example:

Global elements: Large prime number p and a primitive root g of p.

Decryption key: x – private, Calculate $a \equiv g^x \bmod p$, where $x \in \mathbb{Z}$.
Publish (p, g, a).

Encryption:
Let the message is *m*; $(0 < m < p)$ and choose *y* - private $(0 < y < p)$.
Compute $b = g^y \bmod p$. Then,
$c \equiv m.a^y \bmod p$.
Send (*b*, *c*).

k and *m'* are chosen at randomly by the attacker. Note that all are considered in *mod p*.

Let the ciphertext is *C* = (*b*, *c*).
$C = (b, c)$
$= (g^y, m.a^y)$

Now calculate *C'* by the attacker as follows:
C' = ($g^y\ g^k,\ a^y.m.a^k.m'$)

C' = (g^{y+k} , $(m.m').a^{y+k}$))

Give, *C'* to the decryption oracle.

m'' will be return.

Now we can get *m* from *m''*.

C''= (*m. m'*). $\boldsymbol{a^{y+k}}$

m'' = *m. m'*

$m = m''.m'^{-1}$

Therefore we can get the message easily without any keys.

2 PROPOSED IMPROVEMENT OF THE EL-GAMAL PUBLIC KEY CRYPTOSYSTEM

I proposed an improvement for the El-Gamal public key cryptosystem. In this paper, I get the message in numerical form. But, we can get any standard representation for a large message. Consider we have to encrypt a message *m.* In this method, the public encryption key is ($p, g, g^x \bmod p$). Here *p* is any large prime number and g is a primitive root of p . The public encryption key is similar to the public encryption key of the El-Gamal public key cryptosystem.

The structure of the ciphertext *C* has changed on the improvement system. Write *m* = $p_1p_2p_3$ ... p_i ; Where p_i is prime. $(0 < i < P)$. That means we need *i*- prime numbers as products to get *m*. Then $(g^x \bmod p)^y \bmod p$ is multiplied by the number of prime numbers which needs to get *m*. *c* is calculated by the *i* th power of the message *m*. Then we send $(g^y \bmod p,\ m^i, i\ .\ g^{x.y} \bmod p)$.

In decryption process first $g^{xy} \bmod p = (g^x \bmod p)^y$ is calculated. Then $i.(g^x \bmod p)^y \bmod p$. is divided by $(g^x \bmod p)^y$. Now we have *i*. Then taking the inverse of *i* on *c* we can get the message *m*.

You can use this system with following steps.

Step 01: Global elements: Let any large prime number *p* and a primitive root g of p.

Step 02: Decryption key: x – private, Calculate $g^x \bmod p$, where $x \in \mathbb{Z}$.
Publish ($p, g, g^x \bmod p$).

Step 03: Encryption Process:
Let the message is *m*; $(0 < m < p)$ and choose *y* - private $(0 < y < p)$.
Compute $b = g^y \bmod p$.
Write *m* = $p_1p_2p_3$ ... p_i ; Where p_i is prime.
$(0 < i < p)$
Calculate $n = i.a^y \bmod p$
Calculate $c = m^i$
Send (b, c, n)

Step 04: Decryption:
Compute $b^x \bmod p \equiv a^y$. Then,
Calculate $\frac{n}{b^x \bmod p}$
(Note : $\frac{n}{b^x \bmod p} = i$)
$m = c^{1/i}$

2.1 Proof

The extended El-Gamal system decryption expression is $c^{\frac{b^x \bmod p}{n}} = c^{\frac{g^{y.x} \bmod p}{i.a^y \bmod p}} = c^{\frac{g^{y.x} \bmod p}{i.g^{x.y} \bmod p}} = c^{1/i} = (m^i)^{1/i} = m$.

2.2 Procedure

.Let the public key is (g, a, p) and the ciphertext is (b, c, n).
See the figure 01.

2.3 Key Generation for the Extended El-Gamal system

Key generation of the extended El-Gamal system is same as the El-Gamal public key cryptosystem.

Algorithm
Extended_ElGamal_Key_Generation
{
Select a large prime *p*
Select *x* to be a member of the group $(\mathbb{Z}_p^*, \times)$; $1 \le x \le p-2$
Select g to be a primitive root in $(\mathbb{Z}_p^*, \times)$
$a \leftarrow g^x \bmod p$
Public_key ← (g, a, p)
Private_key ← *d*
return Public_key and Private_key
}

2.4 Extended El-Gamal Encryption

Algorithm
Extended_ElGamal_Encryption (g, a, p, i, m)
{
Select a random integer y in the group $(\mathbb{Z}_p^*, \times)$

$b \leftarrow g^y \bmod p$
$n \leftarrow i.a^y \bmod p$
$c \leftarrow m^i$

return b, n and c }

2.5 Extended El-Gamal Decryption

Algorithm

Extended_ElGamal_Decryption (x, p, b, n, c)

{

$m \leftarrow c^{\frac{b^x \bmod p}{n}}$

return m

}

Compute $b \equiv g^y \bmod p \equiv 7^9 \bmod 71 \equiv 47$.

Write $m = 3 * 3 * 3$; Then, $i = 3$

Calculate $n = i.a^y \bmod P \equiv 3 . 41^9 \bmod 71 \equiv 69$

Calculate $c = m^i \equiv 27^3 \equiv 19683$

Send $(b, c, n) = (47, 19683, 69)$

Step 04: Decryption:

Compute $b^x \bmod p \equiv 47^{25} \bmod 71 \equiv 23$. Then,

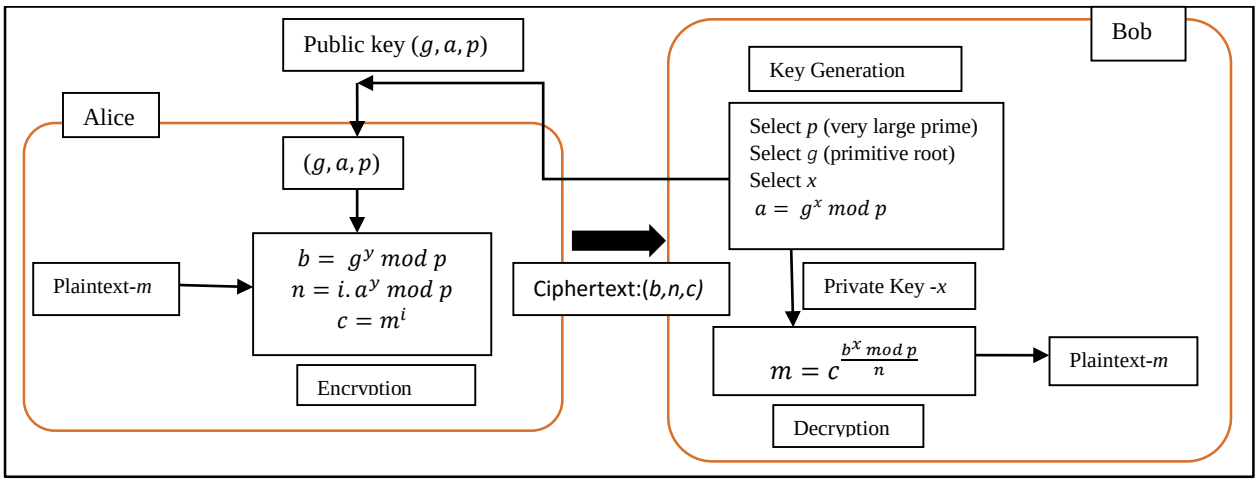

Figure 01- Procedure of the improved system

2.6 Computational complexity

If we use the fast exponential algorithm then encryption and decryption of the extended system can be done in polynomial time.

2.7 Examples for the improved system

Example 01:

Step 01: Select $p = 23$ and a primitive root of $p = 23$ is $g = 5$.

Step 02: Let, $x = 8$.

Calculate $g^x \bmod p \equiv 5^8 \bmod 23 \equiv 16$.

Publish ($23, 5, 16$).

Step 03: Encryption:

Let the message is $m = 6$; and choose $y = 3$.

Compute $a = b \equiv g^y \bmod p \equiv 5^3 \bmod 23 \equiv 10$.

Write $m = 2 * 3$; Then, $i = 2$

Calculate $n = i.a^y \bmod P \equiv 2 . 16^3 \bmod 23 \equiv 4$

Calculate $c = m^i \equiv 6^2 \equiv 36$

Send $(b, c, n) = (10, 36, 4)$

Step 04: Decryption:

Compute $b^x \bmod p \equiv 10^8 \bmod 23 \equiv 2$. Then,

Calculate $\frac{n}{b^x \bmod P} = \frac{4}{2} = 2$

(Note : $\frac{n}{b^x \bmod P} = i$)

$m = c^{1/i} = 36^{1/2} = 6$

Example 02:

Step 01: Select $p = 71$ and a primitive root of $p = 71$ is $g = 7$.

Step 02: Let, $x = 25$.

Calculate $a = g^x \bmod p \equiv 7^{25} \bmod 71 \equiv 41$

Publish ($71, 7, 41$).

Step 03: Encryption:

Let the message is $m = 27$; and choose $y = 9$.

.

Calculate $\frac{n}{b^x \bmod P} = \frac{69}{23} = 3$

(Note : $\frac{n}{b^x \bmod P} = i$)

$m = c^{1/i} = 19683^{1/3} = 27$

3 THE IMMUNITY FOR A CHOSEN CYPHERTEXT ATTACK

Global elements: Let any large prime number p and a primitive root g of p.

Decryption key: x – private, Calculate $g^x \bmod p$, where $x \in \mathbb{Z}$.

Publish ($p, g, g^x \bmod p$).

Encryption Process:

Let the message is m; $(0 < m < p)$ and choose y - private $(0 < y < p)$.

Compute $b = g^y \bmod p$.

Write $m = p_1 p_2 p_3 \ldots p_i$; Where p_i is prime. $(0 < i < p)$

Calculate $n = i.a^y \bmod p$

Calculate $c = m^i$

Send (b, c, n)

Now the attacker gets the ciphertext $C = (b, c, n)$

Attacker chooses values *k, m' and t* randomly. (According to previous attack to the El-Gamal public key cryptosystem, the attacker chooses only two random values. From two values he can never attack to this extended system. So, the attacker chooses 3 values to attack to this extended system).

$C = (b, c, n)$

$= (g^y , m^i , i.a^y \bmod p)$

Now calculate *C'* by the attacker as follows:

$C' = (g^y . g^k \ , \ m^i . m^t \ , \ . a^y \ mod \ p . t . a^k \ mod \ p.)$

$C' = (g^{y+k} \ , \ m^i . m^t \ , \ (i . t) . (a^y \ mod \ p) . (a^k \ mod \ p))$

Give, *C'* to the decryption oracle.

m'' will be return.

$m'' = m^i . m^t$

$m = (\frac{m''}{r^t})^{1/i}$

The attacker does not know the value of i. Therefore he can't get *m* from *m''*.

So, above ciphertext attack will be failure in this extended El-Gamal system.

4 SECURITY OF THE IMPROVED PUBLIC KEY CRYPTOSYSTEM

4.1 Notions of Security

Semantic Security (indistinguishability of Encryptions/ IND): This notion was introduced by Goldwasser and Micali [12]. This property captured the idea according to which an adversary should not be able to get any information about a plaintext, its length excepted given its encryption.

Chosen Plaintext Attack (CPA): The adversary can access an encryption oracle and hence to the encryption of any plaintext.

Non-Adaptive Chosen Ciphertext Attack (CCA1/ Lunchtime Attack/ Midnight Attack): The adversary can access a decryption oracle before being given the challenge ciphertext.

Adaptive Chosen Ciphertext Attack (CCA2): According to Rackoff and Simon [13], the adversary queries the decryption oracle before and after being challenged. But, the adversary may not feed the oracle with the challenge ciphertext itself.

4.2 IND-CPA security of the improved El-Gamal cryptosystem

This improved cryptosystem is IND-CPA secure as IND-CPA security of El-Gamal public key cryptosystem.

Discrete Diffie-Hellman Assumption:

The tuple $(g^x, \ g^y, g^{xy})$ is computationally indistinguishable from $(g^x, \ g^y, g^z)$ for $x, y, z \xleftarrow{\$} \mathbb{Z}_q$.

Theorem: If the Discrete Diffie-Hellman problem is hard then the improved El-Gamal cryptosystem is IND-CPA secure.

Proof: (By contradiction). Assume that an adversary can break the improved El-Gamal cryptosystem, That is, it has significant advantage by a real or random definition,

$AdvA = \Pr[A^{E_{pk}}(pk) = 1] - \Pr[A^{E_{pk\$}}(pk) = 1].$

Since improved cryptosystem is a public key encryption scheme, if it is secure against a single query it is secure against q queries, so we only need to show that it is (t, q, ε) secure for $q = 1$; we can thus assume that the adversary *A* makes exactly one query.

The adversary *A* that runs in time t and has advantage δ, we can construct another adversary *B* for DDH that runs in time $t + O(1)$ and has advantage δ. Algorithm $B(a, b, c)$ is as follows:

1. Run $A^{E_B}(a)$, where *B*'s version of the encryption oracle E_B answers its one query *m* with $(b, c.m)$.
2. Output the same result as *A* does.

In the case where *B* is called on a triple of the form $(g^x, \ g^r, g^{xr})$, what *A* sees is identical to interacting with a real encryption oracle $B(g^x, \ g^r, g^{xr}) = A^{E_{pk}}(pk)$. In the case where B is called on a tuple of the form$(g^x, \ g^r, g^z)$, A sees the values $a = g^x \ and \ (b, c.m) = (g^r, g^z.m)$. Since g^z is selected uniformly at random, $g^z.m$ is also a uniform random value and is thus completely indistinguishable from $g^{zr}x.\$(m)$ and $(g^r, g^z.m)$ is the same distribution as $g^r, g^r x.\$(m)$. This makes *B* a perfect simulator of a random oracle and in this case $B(g^x, \ g^r, g^z) = A^{E_{pk\$}}(pk)$.

This construction thus turns an adversary that breaks Extended El-Gamal cryptosystem into one that breaks DDH with the same advantage, adding constant time complexity.

5 CONCLUSIONS AND FUTURE WORKS

An improvement of El-Gamal public key cryptosystem has presented. The security of this improved system depends on i. If anyone gets i then he can find the message easily. In this system the encryption increases the size of a message. Therefore this improved system is very suitable for small messages or key exchanges.

I try to solve the problem that is the encryption increases the size of a message of above introduced system, using modular exponentiation methods.

6 ACKNOWLADGEMENT

I would like to thank Dr. Sandirigama, M. (Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka), Dr. Ishak , M.I.M. (Department of Engineering Mathematics, Faculty of Engineering, University of Peradeniya, Sri Lanka) and Dr. Alawathugoda, J. (Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka) for their very useful advice in my research work.

7 REFERENCES

[1] [1] Rivest, R., Shamir, A., Adleman, L. 1978. A method for obtaining digital signature and public key cryptosystems. Communications of the ACM, Vol.21 (1978), 120-126.

[2] Diffie, W., Hellman, M. 1976. New directions in Cryptography, IEEE Translations, Information Theory 22 (1976), 644-654.

[3] ElGamal, T. 1985. A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory 31 (1985), 469-472.

[4] Das, A. Public Key Cryptography – Theory and Practice Chapter 3: Algebraic and Number-theoretic Computations, 171-255.

[5] Cramer, R., Shoup. V. 1998. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack, In Crypto '98, Springer-Verlag (1998), LNCS 1462, 13–25.

[6] Naor, M., Yung, M. 1990. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In Proc. of the 22nd STOC, ACM Press (1990), 427–437.

[7] Pointcheval, D. 1999. New Public Key Cryptosystems based on the Dependent-RSA Problem, Advances in

Cryptology – Proceedings of EUROCRYPT ’99, J. Stern Ed. Springer – Verlag, LNCS 1592 (1999), 239-254.

[8] Forouzan, A.B., Mukhopadhyay, D. Cryptography and Network Security, Special Indian Edition, 265-290, 306-316.

[9] Liu, Z., Yang, X., Zhong, W., Han, Y. 2014. An Efficient and Practical Public Key Cryptosystem with CCA-Security on Standard Model, Tsinghua Science and Technology, ISSN 1007-0214 08/13, Vol.19 (2014), 486-495.

[10] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P. 1998. Relations among notions of security for public key encryption schemes, Lecture Notes in Computer Science, vol. 1462 (1998), 26-45.

[11] Tsiounis, Y., Yung, M. 1998. On the security of ElGamal based encryption. In H. Imai and Y. Zheng, editors, Public Key Cryptography, Springer, vol. 1431 of Lecture Notes in Computer Science (1998),117–134.

[12] Goldwasser, S., Micali, S. 1984. Probabilistic Encryption, Journal of Computer and System Sciences 28 (1984), 270-299.

[13] Racko, C., Simon, D.R. 1992. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack, Crypto '91, LNCS 576, Springer-Verlag (1992), 433-444.