GENERALIZATION OF THE ELGAMAL PUBLIC-KEY CRYPTOSYSTEM

RAJITHA RANASINGHE AND PABASARA ATHUKORALA

Abstract. The ElGamal cryptosystem is one of the most widely used public-key cryptosystems that depends on the difficulty of computing the discrete logarithms over finite fields. Over the years, the original system has been modified and altered in order to achieve a higher security and efficiency. In this paper, a generalization for the original ElGamal system is proposed which also relies on the discrete logarithm problem. The encryption process of the scheme is improved such that it depends on the prime factorization of the plaintext. Modular exponentiation is taken twice during the encryption; once with the number of distinct prime factors of the plaintext and then with the secret encryption key. If the plaintext consists of only one distinct prime factor, then the new method is similar to that of the basic ElGamal algorithm. The proposed system preserves the immunity against the Chosen Plaintext Attack (CPA).

Contents

1. Introduction

1.1. **Prologue.** The year 1976 marked a revolution in the theory of Cryptography because of the introduction of the ground breaking concept of public-key cryptography by Whitefield Diffie and Martin Hellman (see [1]). Since then, public-key cryptography which is also known as asymmetric cryptography has been an

Key words and phrases. Public-key cryptography, ElGamal encryption, discrete logarithm problem, Prime factorization, Chosen plaintext attack.

active area of research and also is of significant importance in the real world applications due to its secure key distribution nature in contrast to the symmetric cryptography.

ElGamal public key cryptosystem introduced by Taher ElGamal in 1985 (see [3]) is a probabilistic algorithm that was developed based on the Diffie-Hellman key exchange protocol. Unlike the Diffie-Hellman algorithm, this is a complete encryption-decryption system that depends on the discrete logarithm problem. A probabilistic encryption can be defined as an encryption scheme that generates different ciphertexts when the same plaintext is encrypted several times and the discrete logarithm problem is finding the discrete logarithm to the base g of x in a group G where g is a generator of the group and $x \in G$. The outline of the original ElGamal system is reviewed below.

1.2. **The ElGamal cryptosystem.** Suppose A and B are the two parties involved in a communication over an unsecured channel where A is the receiver and B is the sender.

Key generation: First A chooses a large prime number p and computes a primitive root g modulo p. ElGamal proposes that p must be chosen such that $p-1$ has at least one large prime factor to ensure the security of the system. Let the secret decryption key of A be x $(1 \leq x \leq p-2)$ and the secret encryption key of B be y $(1 \leq y \leq p-2)$. Now A calculates $a \equiv\ g^x (\text{mod } p)$ and publish $p, g,$ and a.

Public keys: p, g, a

Private keys: x, y

Encryption: Now B encrypts the plaintext m as follows.

$$b \equiv\ g^y (\text{mod } p)$$

$$c \equiv\ m \cdot a^y (\text{mod } p)$$

Share $b,$ and c with A.

Decryption: Once receiving b and c, A recovers m.

$$b^x \equiv a^y (\text{mod } p)$$

$$m \equiv c \cdot (\overline{a^y}) (\text{mod} p)$$

where $(\overline{a^y})$ is the inverse of a^y under modulo p.

In [2], Dissanayake has introduced an improvement for the basic ElGamal system using the prime factorization of the plaintext. The message sending structure and the decryption process of the orginal scheme have been changed in order to withstand Chosen Plaintext Attack (CPA) as well as Chosen Ciphertext Attack (CCA).

This paper proposes a generalization for the original ElGamal algorithm incorporating the ideas presented in [2]. The public-key generation and the decryption processes of our scheme is similar to that of the original system. However, the encryption process depends on the prime factorization of the plaintext in addition to the modular exponentiation. Modular exponentiation of the chosen primitive root is taken twice during the encryption; with respect to the number of distinct prime factors of the plaintext and with respect to the secret encryption key.

The new system is proved to be secure against the CPA, given that the Decision Diffie-Hellman (DDH) problem is hard. Note that, for a given group G, a group element g, and the elements g^a, g^b, g^c, determining whether $g^c = g^{ab}$ is defined as the DDH problem. Moreover, each encryption depends on the encryption key as well as on the prime factorization of the plaintext.

The paper consists of two main sections; in section 2 we presents the proposed cryptosystem in detail and in section 3 the strength of the new system against cryptanalysis is presented.

2. Generalization of the ElGamal algorithm

A detailed description of our system is presented in this section. The basic steps of the scheme is explained in general and using a simple example to make the ideas clearer.

2.1. **General Procedure.**

The algorithm is designed under the three primary steps of key generation, encryption, and decryption.

Key Generation: The key generation for the new system is similar to that of the standard ElGamal cryptosystem. First the designer sets the following private and public keys.

Step 1: Select a large prime number p and a primitive root modulo p (Say g).

Step 2: Select the private decryption key x such that $1 < x < p - 1$.

Step 3: Find $a \equiv g^x \pmod{p}$.

Private key: x

Public keys: $p,\ g,\ a$

Encryption: Now the sender obtains the public keys of the intended recipient and encrypt the plaintext as follows.

Step 1: Let the plaintext be m and write m as a product of i distinct prime factors.

$$\text{i.e., } m = {p_1}^{\alpha_1} \cdot {p_2}^{\alpha_2} \cdots {p_i}^{\alpha_i} \ .$$

Step 2: Select the private encryption key y where $1 < y < p - 1$.

Step 3: Calculate the values of $d,\ b,$ and c, Share b and c.

$$\begin{aligned} d &\equiv g^i \pmod{p} \\ b &\equiv d^y \pmod{p} \\ c &\equiv m \cdot a^{iy} \pmod{p} \end{aligned}$$

Ciphertext: b, c

Decryption: Authorized receiving party decrypts the ciphertext using the private decryption key x.

Step 1: Compute b^x in order to determine a^{iy} without the private key of the sender.

$$\begin{aligned} b^x = (d^y)^x &\equiv (g^i)^{xy} \pmod{p} \\ &\equiv (g^x)^{iy} \pmod{p} \\ &\equiv a^{iy} \pmod{p} \end{aligned}$$

Step 2: Recover m by the relation

$$m \equiv c \cdot (\overline{b^x}) \pmod{p}\ ,$$

where $(\overline{b^x})$ is an inverse of b^x under modulo p.

Remark 2.1. Note that if $i = 1$, that is if the plaintext consists of only one distinct prime factor, then our system is equivalent to the original ElGamal algorithm.

2.2. **Example.** For the ease of understanding, now we illustrate our method with an example. Note that, the selection of parameters is done to make the computation simpler, and are not in the usable range for a secure transmission.

Key generation: Let $p = 23,\ g = 5,$ and $x = 8$ respectively be a prime number, a primitive root, and the secret decryption key of the designer, A. Calculate $a \equiv g^x \pmod{p}$:

$$a \equiv 5^8 \pmod{23} \equiv 16 \pmod{23}.$$

Private decryption key: $x = 8$
Public encryption keys: $p = 23,\ g = 5,\ a = 16$

Encryption: Suppose the sender B wants to share the plaintext $m = 6$ with A. Then, B first have to obtain the prime factors of 6; i.e., 2 and 3. So $i = 2$.

Now B chooses his secret encryption key as $y = 3$ and calculates $d, b,$ and c.

$$\begin{aligned} d = g^i &\equiv 5^2 \pmod{23} \\ &\equiv 2 \pmod{23} \end{aligned}$$

$$\begin{aligned} b = d^y &\equiv 2^3 \pmod{23} \\ &\equiv 8 \pmod{23} \end{aligned}$$

$$\begin{aligned} c = m \cdot a^{iy} &\equiv 6 \cdot (16)^{2\cdot 3} \pmod{23} \\ &\equiv 1 \pmod{23} \end{aligned}$$

Finally, B shares the two values $b = 8,$ and $c = 1$ with A.

Decryption: Once the ciphertext values b and c are received, A first calculates b^x;

$$b^x \equiv 8^8 \pmod{23} \equiv 4 (\text{mod } 23).$$

Then the modulo inverse, $(\overline{b^x})$ can be obtained:

$$\begin{aligned} 4 \cdot (\overline{b^x}) &\equiv 1 \pmod{23} \\ (\overline{b^x}) &\equiv 6 \pmod{23}. \end{aligned}$$

Thus A can recover the plaintext as follows:

$$\begin{aligned} m &\equiv c \cdot (\overline{b^x}) \pmod{p} \\ &\equiv 1 \cdot 6 \pmod{23} \equiv 6 \pmod{23}. \end{aligned}$$

3. Cryptanalytic approach

The cryptosystem we have introduced is IND-CPA secure, which is similar to the IND-CPA security of the ElGamal algorithm, where IND-CPA stands for the *Indistinguishability under Chosen Plaintext Attack.* According to [4], indistinguishability implies that an adversary cannot distinguish the encryption of any two plaintexts m_0 and m_1, chosen by the adversary, of the same length. Moreover, in a chosen plaintext attack, the adversary is given access to an encryption oracle.

Before proving the CPA security of the system, few useful results are discussed below.

Definition 3.1. ([5])

A public key encryption scheme $\pi = (\text{Gen, Enc, Dec})$ has indistinguishable encryptions under chosen plaintext attacks (or is CPA secure) if for all probabilistic, polynomial-time adversaries $\mathcal{A}$, there exists a negligible function "negl" such that,

$$\Pr[\text{Pub } \mathcal{K}^{\text{cpa}}_{\mathcal{A},\pi}(n) = 1] \leq \frac{1}{2} + \text{negl}(n) \ .$$

The CPA security of the ElGamal encryption is based on the hardness of the Decision Deffie-Hellman (DDH) problem.

Let $\mathcal{G}$ be a polynomial-time algorithm that on input 1^n, outputs a cyclic group $\mathbb{G}$, its order p (with $\|p\| = n$), and a generator g, where the group operation in $\mathbb{G}$ can be computed in polynomial-time n. Then,

Definition 3.2. [5]

We say the DDH problem is hard relative to $\mathcal{G}$ if for all probabilistic, polynomial-time algorithms $\mathcal{A}$, there exists a negligible function negl such that

$$|\ \Pr[\mathcal{A}(\mathbb{G}, p, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, p, g, g^x, g^y, g^{xy}) = 1]\ | \leq \text{negl}(n),$$

where in each case, the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs $(\mathbb{G}, p, g)$ and the random $x, y, z \in \mathbb{Z}_p$ are chosen.

The security of the ElGamal algorithm is proved by the following theorem.

Theorem 3.3. [5]

If the DDH problem is hard relative to $\mathcal{G}$, then the ElGamal encryption scheme has indistinguishable encryptions under chosen plaintext attacks.

This theorem can be applied to prove the CPA security of the introducing generalization with an analogous proof. Thus, we will prove the theorem for our cryptosystem.

Theorem 3.4. *If the DDH problem is hard relative to $\mathcal{G}$, then the generalization of the ElGamal encryption scheme has indistinguishable encryptions under chosen plaintext attacks.*

Proof. Consider $\mathcal{A}$ to be an adversary against the CPA-security of the scheme and $\mathcal{B}$ to be an adversary for DDH.

$\mathcal{B}$ is called upon a DDH instance $(\mathbb{G}, p, g, g^x, g^{iy}, h)$ where either $h = g^{iyx}$ (if $b = 0$) or $h = g^z$ for a uniform random $z \in \mathbb{Z}_p$ (if $b = 1$).

$\mathcal{B}$ shares the public keys $(\mathbb{G}, p, g, g^x)$ with $\mathcal{A}$.

Now $\mathcal{A}$ outputs the two plaintexts m_0 and m_1. Without loss of generality, assume that $m_0 = 0$ and $m_1 = 1$ where $b \in \{0, 1\}$.

After encrypting m_0 and m_1, $\mathcal{B}$ sends the challenge ciphertext c^* to $\mathcal{A}$ where

$$c^* = \mathrm{E}(m_b) = (g^{iy}, h)$$

.

If $b = 0$, then c^* is the proper encryption of the plaintext m_0. However, if $b = 1$, then c^* is not an actual encryption scheme since there is no way for the receiver to decrypt.

Finally, upon receiving $\mathcal{A}$'s guess for b (say b'), $\mathcal{B}$ outputs the value b'.

Assuming DDH problem is hard relative to $\mathcal{G}$, by Definition 2 we get the relation

$$\begin{aligned}
\mathrm{negl}(n) \geq & \mid \Pr[\mathcal{B}(\mathbb{G}, p, g, g^x, g^{iy}, g^{iyx}) = 1] - \Pr[\mathcal{B}(\mathbb{G}, p, g, g^x, g^{iy}, g^z) = 1] \mid \\
= & \mid 1 - \Pr[\mathcal{B}(\mathbb{G}, p, g, g^x, g^{iy}, g^{iyx}) = 0] - \Pr[\mathcal{B}(\mathbb{G}, p, g, g^x, g^{iy}, g^z) = 1] \mid \\
= & \mid 1 - \Pr[\text{Pub } \mathcal{K}^{\mathrm{cpa}}_{\mathcal{A},\pi}(n) = 1 \mid b = 0] - \Pr[\text{Pub } \mathcal{K}^{\mathrm{cpa}}_{\mathcal{A},\pi}(n) = 1 \mid b = 1] \mid \\
= & \mid 1 - 2\Pr[\text{Pub } \mathcal{K}^{\mathrm{cpa}}_{\mathcal{A},\pi}(n) = 1] \mid
\end{aligned}$$

for a negligible function 'negl'.

Thus, we can obtain

$$\Pr[\text{Pub } \mathcal{K}^{\mathrm{cpa}}_{\mathcal{A},\pi}(n) = 1] \leq \frac{1}{2} + \mathrm{negl}(n)$$

.

That is, the scheme is CPA secure by Definition 1.

□

4. Conclusion and Future work

In this paper we have proposed a generalization for the ElGamal public-key cryptosystem whose security depends on the discrete logarithm problem. The encryption process of the new scheme rests upon the prime factorization of the plaintext in addition to the modular exponentiation. Modular exponentiation increases the size of the plaintext. Thus, the system presented is more suitable for shorter messages or symmetric key distribution.

Our system is proved to be secure under chosen plaintext attack. However, the security issues associated with the system needs to be further examined in more detail. Developing the scheme to withstand possible cryptanalysis is our immediate next task.

References

[1] Whitfield D. and M. Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[2] W.D.M.G.M Dissanayake. An improvement of the basic elgama public key cryptosystem. *International Journal of Computer Applications Technology and Research*, 7(2):40–44, 2018.

[3] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[4] Steven D Galbraith. *Mathematics of public key cryptography*. Cambridge University Press, 2012.

[5] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.