Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 3, pp. 631–639
DOI : 10.47974/JDMSC-1723

A modified ElGamal-like cryptosystem

Rajitha Ranasinghe *
Pabasara Athukorala †
Pramodya Athurugiriya §
Department of Mathematics
University of Peradeniya
Peradeniya 20400
Sri Lanka

Abstract

The ElGamal cryptosystem stands as one of the foremost public key encryption methods and represents a probabilistic algorithm, originating from the framework of the Diffie-Hellman key exchange protocol. Diverging from the Diffie-Hellman approach, ElGamal cryptosystem constitutes a comprehensive encryption-decryption scheme hinging on the discrete logarithm problem for its security. Its robustness is rooted in the formidable challenge of computing the discrete logarithm modulo a substantially large prime number. In this study, a variation of the ElGamal algorithm was introduced using matrices. The key generation of the new system is different from that of the standard ElGamal Cryptosystem. Here, our generator is a 2×2 matrix containing four primitive roots of the selected prime number. The Plaintext is considered in uppercase letters and then converted into uppercase and lowercase letters alternatively. ASCII alphabet is used to obtain the corresponding numerical values for each letter and decomposed into blocks of 2×2 matrices. The security of this system rests on the intrinsic complexity of the discrete logarithm problem, a computational challenge well-documented for its arduousness. Additionally, our proposed system maintains resilience against Chosen Plaintext Attacks (CPA), further bolstering its security posture.

Subject Classification (2010): *11Z05, 11A07, 11T71, 94A60.*

Keywords: *ASCII alphabet, Chosen plaintext attack, Diffie-hellman key exchange, ElGamal cryptosystem.*

* *E-mail:* **rajithamath@sci.pdn.ac.lk (Corresponding Author)**
† *E-mail:* **pabasaraat@sci.pdn.ac.lk**
§ *E-mail:* **pramodyasachinthani@gmail.com**

1. Introduction

1.1 *Prologue*

Cryptography encompasses the examination of mathematical techniques aimed at safeguarding digital data, systems, and distributed computations from hostile incursions. Public key cryptography or asymmetric cryptography, distinguishes itself by employing a pair of distinct keys, as opposed to private key or symmetric encryption, where a single key suffices for the task. The notion of public key cryptography was first introduced by Martin Hellman, Ralph Merkle, and Whitfield Diffie in 1976 [1]. Subsequently, public key cryptography has evolved into a vibrant realm of research and has gained substantial significance in practical applications owing to its secure key distribution capabilities. The ElGamal cryptosystem introduced by Taher ElGamal in 1985 [2] is a public key cryptosystem that is based on the Diffie-Hellman key exchange protocol and hence, its security depends on the difficulty of computing the discrete logarithms over finite fields. Throughout the years, the initial system has undergone adaptations and refinements, with the aim of enhancing both security and efficiency. For more contemporary research within this domain, interested readers may refer [3] which proposes and analyzes an algorithm called the Diffie-Hellman Text-to-Image Encryption (DHTTIE) algorithm which is based on the much celebrated Diffie-Hellman algorithm and the Text-to-Image Encryption (TTIE) algorithm.

Also, [4] proposes a generalized version of the ElGamal cryptosystem that enhances security by incorporating the prime factorization of the plaintext into the encryption process. The scheme applies modular exponentiation twice and retains resistance to Chosen Plaintext Attacks while remaining dependent on the discrete logarithm problem.

1.2 *The Conventional ElGamal Cryptosystem*

Consider a hypothetical scenario in which X serves as the recipient, and Y as the sender during a communication over an unsecured channel. The analysis will be explained considering the three main aspects; Key Generation, Encryption, and Decryption.

Key Generation

X begins the communication by selecting a relatively larger prime number p, ensuring that $p-1$ has at least one large prime factor to enhance

the security of the cryptosystem. She then computes a primitive root g modulo p.

We shall denote X's secret decryption key by k, where $1 \le k \le p-2$, while Y's decryption key by m, where $1 \le m \le p-2$.

X computes $a \equiv g^k \, (mod\ p)$ and publishes the values p, g, and a. Consequently, the public keys are p, g, a, while the private keys are k and m.

Encryption

Now Y encrypts the plaintext s using

$$b \equiv g^m (\text{mod } p) \text{ and } c \equiv s \cdot a^m (\text{mod } p)$$

and share b and c with X.

Decryption

After b and c are received, X recovers s by computing

$$b^k \equiv a^y (\text{mod } p) \text{ and } s \equiv c \cdot (\overline{a^m}) (\text{mod } p),$$

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

The proposed variation of the ElGamal algorithm closely follows the conventional system in terms of public-key generation and decryption processes. However, the key generation phase introduces a notable difference. Instead of a single generator g, our system employs a 2 × 2 matrix containing four primitive roots of the chosen prime number p. To ensure the existence of four primitive roots, the selected prime p should be at least 11.

The plaintext is represented using uppercase letters, which are then alternately converted into uppercase and lowercase letters. By leveraging the ASCII alphabet, each letter is assigned a corresponding numerical value, and these values are subsequently arranged into 2 × 2 matrices.

This matrix-based approach maintains the fundamental security principles of the ElGamal cryptosystem while introducing a new structural element that may offer additional benefits in specific applications. The security of this modified system remains rooted in the inherent complexity of the discrete logarithm problem, a well-established challenge in cryptographic literature.

Given a cyclic group G with a generator g, consider the elements g^a, g^b, and g^c. The question is whether $g^c = g^{ab}$ is known as the Decision Diffie-Hellman (DDH) problem. The security of the present system is

underpinned by the assumption that the DDH problem is computationally difficult to solve. As a result, our proposed cryptographic scheme has been shown to be resilient against Chosen-Plaintext Attacks (CPA), further reinforcing its robustness in practical applications.

2. The Proposed Algorithm

This section provides an exhaustive overview of our system. Section 2.1 delves into the fundamental mathematical findings that underpin our algorithm. Following this, subsequent sections outline the core steps of the scheme, presented in a general context and supported by a straightforward example to enhance clarity.

2.1 *General Procedure*

The algorithm has been structured around three primary stages: key generation, encryption, and decryption.

Key Generation

First, the private and public keys are chosen, Private key: a and Public keys: p, g, A.

Step 1: Choose a prime number p and select four primitive roots of modulo p, denoted by r_i for $i = 1, 2, 3, 4$. These are arranged to form a 2 × 2 matrix g:

$$g = \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}$$

is invertible.

Step 2: Select a such that $1 \leq a \leq p - 1$ as the private decryption key.

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

Encryption:

The sender then obtains the recipient's public key and encrypts the plaintext using the following method.

Step 1: Select a plaintext, ignore space between words and convert it into uppercase and lowercase letters alternatively. By using the ASCII alphabet, we can obtain the corresponding numerical values for each letter and form

blocks such that each containing four elements. We used a lowercase letter x to complete the four elements contained in a block. Then convert it into a 2 × 2 matrix. i.e.,

$$M \equiv \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} (\text{mod } p).$$

Step 2: Select b such that $1 \le b \le p-1$ as the private encryption key.

Step 3: Find $B \equiv g^b \pmod{p}$.

Step 4: Calculate the values B_1 and C, satisfying the two congruences:

$$B_1 \equiv A^b \pmod{p} \text{ and } C \equiv M \cdot B_1 \pmod{p}.$$

Now, share the ciphertext C.

Decryption:

Step 1: Compute $A_1 \equiv B^a \pmod{p}$.

Step 2: Recover M using the congruence,

$$M \equiv \overline{A_1} \cdot C \pmod{p},$$

where $\overline{A_1}$ is an inverse of A_1 modulo p.

2.2 *Example*

To enhance the understanding of the present algorithm, we provide an example. It's important to clarify that these parameters were selected for simplifying calculations and are not suitable for secure transmissions.

Key generation:

Let $p = 127$ be the prime number, $g = \begin{bmatrix} 3 & 6 \\ 7 & 12 \end{bmatrix}$ be the matrix whose entries are the primitive roots of 127, and $a = 2$ be the secret decryption key of X.

Calculate $A \equiv g^a \pmod{p}$:

$$A \equiv \begin{bmatrix} 3 & 6 \\ 7 & 12 \end{bmatrix}^2 \equiv \begin{bmatrix} 51 & 90 \\ 105 & 59 \end{bmatrix} (\text{mod } 127)$$

Private decryption key: $a = 2$

Public encryption keys: $p = 127$, $g = \begin{bmatrix} 3 & 6 \\ 7 & 12 \end{bmatrix}$, $A = \begin{bmatrix} 51 & 90 \\ 105 & 59 \end{bmatrix}$.

Encryption:

Suppose Y wants to share the plaintext,

HAVE A NICE DAY!

with X. First, convert it into,

HaVe A nIcE dAy!

and makes blocks as each block contains four letters without considering the spaces.

By using the ASCII alphabet,

H	a	V	e	A	n	I	c	E	d	A	y	!
72	97	86	101	65	110	73	99	69	100	65	121	33

Take $M = \begin{bmatrix} 72 & 97 \\ 86 & 101 \end{bmatrix}$. Now Y chooses $b = 3$ as the secret encryption key and calculates B, B_1, and C.

$$B \equiv \begin{bmatrix} 3 & 6 \\ 7 & 12 \end{bmatrix}^3 \equiv \begin{bmatrix} 21 & 116 \\ 93 & 68 \end{bmatrix} (\text{mod } 127)$$

$$B_1 \equiv \begin{bmatrix} 51 & 90 \\ 105 & 59 \end{bmatrix}^3 \equiv \begin{bmatrix} 53 & 37 \\ 22 & 45 \end{bmatrix} (\text{mod } 127)$$

$$C \equiv M \cdot B_1 \equiv \begin{bmatrix} 108 & 44 \\ 49 & 107 \end{bmatrix} (\text{mod } 127).$$

Finally, Y shares the values B and C with X.

Decryption:

Upon receiving th ciphertext values B and C, X computes the plaintext.

$$A_1 \equiv \begin{bmatrix} 21 & 116 \\ 93 & 68 \end{bmatrix}^2 \equiv \begin{bmatrix} 53 & 37 \\ 22 & 45 \end{bmatrix} (\text{mod } 127)$$

$$D \equiv \overline{A_1} \cdot C \equiv \begin{bmatrix} 72 & 97 \\ 86 & 101 \end{bmatrix} (\text{mod } 127) \equiv M$$

3. Cryptanalytic Approach

In this section, we prove that the proposed cryptosystem is CPA secure. We shall adhere to the notations used in [5]).

The following result is used to prove the CPA security.

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

The CPA security of the proposed ElGamal cryptosystem is established as follows. If the proposed system satisfies the Theorem, then the new system preserves security against the CPA.

Proof: Let $\mathcal{A}$ be a probabilistic polynomial time (PPT) adversary and if the things were truly random, consider $\overline{\Pi}$, where Gen is same as Π, but encryption of a message m with respect to pk = $< \mathbb{G}, q, g, g^a, h >$ is performed by choosing a random $b \leftarrow \mathbb{Z}_q$ and $c \leftarrow \mathbb{Z}_q$ then output is $< g^b, g^c \cdot m >$.

Since g^c is a uniformly distributed group element that is independent of m and g^b also independent of m, the entire ciphertext contains no information on m. This implies that,

$$\Pr[Pub\mathcal{K}^{eav}_{\mathcal{A},\overline{\pi}}(n) = 1] = \frac{1}{2}.$$

Let D be an algorithm whose goal is to solve the DDH problem, where $h_3 = g^{ab}$ is or g^c, with a, b, and c chosen uniformly at random. We consider two cases for analysis.

Case 1: Let the input of D be generated by running $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$ by choosing a random $a, b, c \leftarrow \mathbb{Z}_{\|}$ and finally setting $h_1 = g^a$, $h_2 = g^b$ and $h_3 = g^c$. Next D runs $\mathcal{A}$ on pk where pk = $< \mathbb{G}, q, g, g^a >$ and ciphertext is obtained as $< C_1, C_2 > = < g^b, g^c \cdot m_t >$.

In this case the view of $\mathcal{A}$ when run as a subroutine by D is distributed identically to $\mathcal{A}'s$ view in experiment $Pub\mathcal{K}^{eav}_{\mathcal{A},\overline{\pi}}(n)$.

Algorithm D outputs 1 exactly when the output t_1 of $\mathcal{A}$ is equal to t.

This implies that,

$$\Pr[D(\mathbb{G}, q, g, g^a, g^b, g^c) = 1] = \Pr[Pub\mathcal{K}^{eav}_{\mathcal{A},\overline{\pi}}(n) = 1] = \frac{1}{2}.$$

Case 2: Let the input of D is generated by running $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$ by choosing random $a, b \leftarrow \mathbb{Z}_q$ and setting $h_1 = g^a$, $h_2 = g^b$, and $h_3 = g^{ab}$. Next D runs $\mathcal{A}$ on pk, where pk = $\langle \mathcal{A}, q, g, g^a \rangle$ and ciphertext obtained as $\langle C_1, C_2 \rangle = \langle g^b, g^{ab} \cdot m_t \rangle$.

In this scenario, the distribution of $\mathcal{A}'s$ view when executed as a subroutine by D is identical to its view in the experiment $Pub\mathcal{K}^{eav}_{\mathcal{A},\pi}(n)$.

Algorithm D outputs 1 exactly when the output t_1 of $\mathcal{A}$ is equal to t. This implies that,

$$\Pr[D(\mathbb{G},q,g,g^a,g^b,g^{ab})=1]=\Pr[Pub\mathcal{K}^{eav}_{\mathcal{A},\pi}(n)=1].$$

Given the computational hardness of the DDH problem in $\mathbb{G}$, by Definition 3.9 in [5], a function `negl', exists such that

$$|\Pr[D(\mathbb{G},q,g,g^a,g^b,g^c)=1]-\Pr[D(\mathbb{G},q,g,g^a,g^b,g^{ab})=1]\ |\leq \text{negl}(n)$$

$$\left|\frac{1}{2}-\Pr[Pub\mathcal{K}^{eav}_{\mathcal{A},\pi}(n)=1]\right|\leq \text{negl}(n)$$

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

This completes our proof of establishing that our system is CPA secured.

4. Conclusions and Future Directions

Our present study introduces an ElGamal-like algorithm employing matrix theory, leveraging the computational complexity of the discrete logarithm problem to ensure system security. Our method withstands chosen plaintext attacks, enhancing its robustness. Implementation involves 3×3 matrices featuring randomly chosen primitive roots under modulo p. Nevertheless, efficiency assessment and in-depth security scrutiny are essential for future work. Our next objective centers on exploring the efficiency of the system and delving into interrelated security concerns. Additionally, we aim to bolster the system against potential cryptanalytic attacks, addressing emerging vulnerabilities. This holistic approach underscores the need for comprehensive evaluation and continuous development to establish a robust and resilient cryptographic scheme.

References

[1] W. Diffie and M. Hellman, "New directions in Cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644-654 (1976).

[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on Discrete logarithms," *IEEE transactions on information theory,* vol. 31, no. 4, pp. 469-472 (1985).

[3] A. Abusukhon, M. N. Anwar, Z. Mohammad, and B. Alghannam, "A hybrid network security algorithm based on diffie hellman and text-to-image encryption algorithm," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 22, no. 1, pp. 65-81 (2019).

[4] R. Ranasinghe and P. Athukorala, "A generalization of the Elgamal public-key cryptosystem," *Journal of Discrete Mathematical Sciences and Cryptography* (2021) (Published Online, URL : https://doi.org/10.1080/09720529.2020.1857902).

[5] J. Katz and Y. Lindell, Introduction to Modern Cryptography. CRC Press (2021).

Received May, 2021

Revised September, 2021