Modified Elgamal Cryptosystem Algorithm : A Practical Implementation

Neha Patel¹, Sakshi Sharma², Shivanshu Gupta³
GLA Institute of Technology, Mathura, India
GLNA Institute of Technology, Mathura, India

Abstract—The project Modified ElGamal Cryptosystem Algorithm divides the private key into n+1 parts that is odd number of parts so that even if the intruder is able to detect some of the parts of the private key, it is computationally infeasible to detect all the n+1 parts. At present, the Elgamal encryption algorithm works by sending data to the receiver who has just one private key to decrypt the data. But in this algorithm, as there is just one private key, it can be guessed by any intruder and is thus not reliable. With the need to protect the integrity and privacy of information belonging to individuals and organizations, we have developed this system, this topic was chosen so as to unable the intruder identify the message even if he succeeds in calculating one of the private keys because the calculation of all n+1 parts of the private key is a difficult task. This project aims at converting the plaintext into a form unreadable by unauthorized people and hence can be readily transferred across the web and decrypted at the recipient side only by authorized people. It provides an interactive environment to encrypt, decrypt or transfer encrypted files without compromising with the integrity and privacy of critical information. In the era of wide area, open distributed systems, this system will help resolve various security issues.

I. INTRODUCTION

Security measures must be incorporated into computer systems whenever they are potential targets for malicious or mischievous attacks. This is especially for systems which handle financial transactions or confidential, classified or other information whose secrecy and integrity are critical. With the need to protect the integrity and privacy of information belonging to individuals and organizations, we have developed this algorithm.

1.1 The Technology Behind Cryptography:-

Cryptography is probably the most important aspect of communication’s security and is becoming increasingly important as a basic building block for computer security. The increased use of computer and communications system by industry has increased the risk of theft of proprietary information although these threats may require a variety of counter measures. Encryption is a primary method of protecting valuable electronic information. Encryption is the process of encoding a message in such a way as to hide its contents. Modern cryptography includes several secure algorithms for encrypting and decrypting messages. They are all based on the use of secrets called keys. A cryptography key is a parameter used in an encryption algorithm in such a way that the encryption can not be reversed without the knowledge of the key. Terms used in cryptography are as follows:

- **Plain text:** original message is known as plain text.
- **Cipher text:** coded message is known as cipher text.
- **Encryption:** the process of converting the plain text to cipher text is known as encryption.
- **Decryption:** the process of restoring the plain text from the cipher text is known as decryption.

1.2 Existing System and Problem Statement:-

In this project we are modifying the existing Elgamal encryption algorithm by dividing the private key and assigning them to n+1 authorized receivers individually. The persons will be able to decrypt the message received from the sender only if they are together, separately this operation being impossible for them. It has the following operations:
Key generation: The receiver who wishes to get message, chooses a large prime number \( p \), a random number \( g \) which is also prime and less than the prime number initially chosen and a random integer \( x \) from 0 to \( (p-1) \). He then calculates

\[
y = (r_1)^x \mod p_1
\]

The public key of the sender is \((p, g, y)\) and his private key is \(x\).

Encryption by the sender: The sender generates an integer \( k \) lying between 0 to \((p-1)\). He then calculates

\[
s = (r_1)^k \mod p_1\quad \text{and} \quad t = (y^k \cdot M) \mod p_1
\]

and transmits \((r, t)\) as the encrypted message.

Decryption of the ciphertext: The receiver with his private key calculates

\[
t \cdot s^x \mod p_1
\]

Which gives the plaintext. But in this algorithm, as there is just one private key, it can be guessed by any intruder and is thus not reliable.

1.3 Algorithm:-

- A large prime number \( p \) and a random number \( r \) which is prime and less than the initially chosen prime number is chosen.
- Then after from \( \{0,\ldots,p-1\} \) there are chosen the elements \( x_1, x_2, \ldots, x_{n+1} \), preferably distinct.
- Calculate

\[
y_1 = (r_1)^{x_1} \mod (p), y_2 = (r_1)^{x_2} \mod (p), \ldots, y_{n+1} = (r_1)^{x_{n+1}} \mod (p).
\]

The public key is \(\{p, r, y_1, y_2, \ldots, y_{n+1}\}\) and the private key consists of \(\{x_1, x_2, \ldots, x_{n+1}\}\).

- The sender encrypts message \( m \) knowing the public key as: choose a random element \( z \) from \(\{0,\ldots, p-1\}\) and calculates \( e_1 = (r) \mod p, e_2 = m \cdot y_1 \mod p, \ldots, e_{2n+1} = m \cdot y_{n+1} \mod p \). Then sends the encrypted message \((e_1, e_2)\) to the recipient.

To decrypt the message \((e_1, e_2)\), calculate

\[
x_1 \cdot (e_1^{x_2} \mod (p)), e_1^{x_2} \mod (p) = m.
\]

1.4 EXAMPLE

Let \( p = 7, r = 3, x_1 = 5, x_2 = 9, x_3 = 7, m = 10 (\text{message}) \)

\[
y_1 = (3)^5 \mod 7 = 5
\]

\[
y_2 = (3)^9 \mod 7 = 6
\]

\[
y_3 = (3)^7 \mod 7 = 3
\]

Public Key: \(\{7, 3, 5, 6, 3\}\)

Private Key: \(\{5, 9, 7\}\)

Encryption

Let some choose a random number as \( z = 3 \)

\[
e_1 = 3 \mod 7 = 3
\]

\[
e_2 = (10)(5)(\mod 7) = 1
\]

\[
e_3 = (10)(6)(\mod 7) = 4
\]

\[
e_4 = (10)(3)(\mod 7) = 2
\]

\[
e_5 = e_2 \cdot e_3 \cdot (\mod p) = 1.2.4(\mod 7)z = 1.3 = 3
\]

So Encrypted message is: \((e_1, e_2) = (3, 3)\)

Decryption

\[
x_1 \cdot (e_1^{x_2} \mod (7)) \cdot e_1^{x_3} \mod (7) = (e_1 \cdot e_2 \cdot e_3 \mod (7)) = 10 = m
\]

II. WORKFLOW

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Fig 2.1 Sequence Diagram

2.1 Use Case Diagram:-

It abbreviates to unified modeling language. It includes the detail design of the project.
III. IMPLEMENTATION

Choose on which environment we want to run our code.

Random Key Generation Window:

Window Showing The Prime Number And Random Number Generated:

Fig 2.2: Use Case Diagram

Fig 2.3: Environment To Run

Fig 2.4: Key Generation

Random Number Generation:

Fig 2.5: Prime number Generation

Fig 2.6: Random Number Generation

Window For Getting The Private Key Values From The User:

Fig 2.7: Private Key Generation
Public Key Generation

File Chooser Dialogue Box When Encrypting

Fig 2.8: Public Key Generation

Fig 2.9: Encryption

Fig 3.1: File is being encrypted

Give the location of the decrypted file

Fig 3.2: Location of Decrypted File

Decrypted file is ready

Fig 3.3: Environment To Run

IV. CONCLUSION

We require a project that is easy to understand and easy to implement. For this purpose we started our work right from the scratch. We first went for an extensive survey which provided us with a healthy idea about the need and implementation requirements of the product i.e. about people’s choice, their involvement in these activities etc. This survey really helped us in getting into the majority’s mind and working according to it. This step is followed by methods and approaches which we had to adopt to carry on our workings in a rather smooth way. For this we have chosen waterfall model.

REFERENCES