Enhancing the Security in ElGamal Cryptosystem using Paring Functions

Mani K., Barakath Begam A.

Abstract: The potential disadvantage of ElGamal cryptosystem is the ciphertext produced is always twice as long as the plaintext i.e., the message expansion by a factor of two takes place during encryption. When the message is too long the ciphertext produced by the ElGamal cryptosystem is also too long, i.e., when the ciphertexts are transmitted through the communication channel which lead to provide less security because if anyone of the ciphertext from two ciphertexts for each character of the plaintext is intercepted by the adversary, the other may be retrieved easily because there is a relationship between the two ciphertexts. If two ciphertexts are reduced to one, the adversary may not be able to predict the two ciphertexts from one. To enhance the security of ElGamal cryptosystem, the binary Cantor function \( c: N \times N \to N \), Rosenberg pairing function and Elegant pairing functions are used in this paper. When the said functions are used, the two ciphertexts produced by each plaintext character are reduced to one, so that the adversary cannot easily be recovered the plaintext. Experimental results clearly revealed enhancing the security of ElGamal cryptosystem after incorporating the pairing functions into it.

Keywords: ElGamal Cryptosystem, Cantor Pairing, Rosenberg pairing function, Elegant pairing function, Security.

1. INTRODUCTION

ElGamal cryptosystem is a kind of public key cryptosystem which utilizes randomization in the encryption process. It was proposed by Taher ElGamal in 1984 [1]. It is used in the free GNU Privacy Guard System, recent version of PGP. The security of ElGamal is based on the intractability of the Discrete Logarithm Problem (DLP) i.e., \( Z^*_p \) in the group \( G \) and the Diffie-Hellman Problem (DHP). The \( G \) should satisfy the two important conditions viz., (i) the \( G \) should be relatively easy to apply for efficiency and (ii) the \( G \) should be computationally infeasible because of DLP for security. This is because the ElGamal encryption scheme can be viewed as simply comprising a DH key exchange to determine a session key \( (a^x)^y \), and then encrypting the message by multiplication with that session key [2]. It is a non-deterministic-cryptosystem because the random key \( K \) is chosen each time which results in for the same plaintext \( (M) \) multiple times produce different ciphertext \( (C) \). The padding scheme and the properties of underlying \( G \) play a vital role in determining the security of ElGamal cryptosystem. It is noted that ElGamal cryptosystem produces double \( c \in C \) for each \( m \in M \). Suppose, the length of the \( M \) is too long, the \( C \) obtained from the \( M \) using ElGamal cryptosystem is also too long.

When it is transmitted through the communication channel, it will take more time which leads to customer dissatisfaction and impatience. In order to avoid it, pairing is performed before and after applying the ElGamal encryption. For that, three well known pairing schemes viz., (i) Cantor pairing (ii) Elegant pairing and (iii) Rosenberg pairing are taken in this paper.

The rest of the paper is organized as follows. The various works related to different pairing schemes are discussed in section 2. Mathematical preliminaries related to pairing functions are presented in section 3. Section 4 describes the proposed methodology of ElGamal encryption with pairing schemes. The result of the proposed methodology is discussed in section 5. Finally, section 6 ends with conclusion.

II. RELATED WORKS

Arnold L. Rosenberg [3], provided a short tour about the pairing functions for computation situations and then they discussed about computationally simplest pairing functions-Cauchy–Cantor “diagonal” polynomial. They described the usage of pairing functions based on two specific computational situations in the paper viz., (i) in storage mappings for rectangular arrays/tables, pairing functions could be used for expand and shrink them dynamically and (ii) pairing function was used for instilling accountability mechanism into web-computing projects.

In [4], R. Reddiah discussed a different types of pairing functions viz., Cantor pairing, Elegant pairing, Tate Pairing, Weil pairing, which have a unique nature of handling real numbers \( (N) \) while processing the data. The advantages and disadvantages of pairing functions were also discussed in this paper.

B. Harikrishna et al. [5] proposed a symmetric-key encryption method for encrypting two messages at the same time using pairing function which reduced the original length of the \( M \) into half. Using the transposition cipher, the \( K \) was embedded within the \( C \). Then, it was decrypted with the help of depairing function which produce the original \( M \). The problem of squeezing multiple ciphertexts without losing original information of \( M \) was dealt by Myungsun Kim in [6]. For that purpose, they formalized the notion of decomposability for public-key encryption and investigated the reason for adding the decomposability of challenging. They constructed an ElGamal encryption scheme over extension field and illustrated the support for efficient decomposition. They analyzed the security scheme under the standard DH assumption and evaluated the program too. Steven D. Galbraith et al. [7], aimed to outline the basic choices and usages of pairing functions available for cryptographic schemes in a simple way.
They summarized the main properties and efficiency issues of pairing functions. The work intended the non-specialists to make use of the pairing, who are interested in using pairings for designing the cryptographic schemes. In [8], Mehmet Sabir Kiraz and Osmanbey Uzunkol mainly focused and gave the correct use of the pairing based cryptography in an understandable, informative and most up-to-date manner and they gave a compact and a correct usage of the pairings.

In [9], an identity-based encryption from Weil pairing was proposed. A variant of the computational DHP was assumed for choosing the C security in the random oracle model. The proposed scheme was based on the bilinear maps between the groups. They used the Weil pairing on elliptic curves. For security, the identity based encryption schemes was précised the definitions and also they gave several applications.

Neal Koblitz and Alfred Menezes [10], examined the need of security impact based on pairing function in cryptosystem. At first they described three reasons, why the users may have the long-term viability concerned about high-security of those systems. Then, two families of elliptic curves were discussed for pairing. At first, the pairing values over the prime field \( p \) has been taken from the defined curve. Now, the super singular curves with embedding degree \( k = 2 \) were exists in this family. Finally, the efficiency of the Weil pairing opposed to the Tate pairing was précised the definitions and also they gave a compact and a correct usage of the pairing based cryptography in an understandable, informative and most up-to-date manner and they gave several applications.

III. MATHEMATICAL PRELIMINARIES

This section describes the various pairing schemes viz., Cantor, Rosenberg’s and Elegant pairing functions which are useful to pair the numbers in this paper.

Definition 1: Pairing Function

It is a bijection between \( N \times N \) and \( N \) that is also strictly monotone in each of its arguments.

\[ p : N \times N \rightarrow N \] (1)

where \( N \) is a natural number.

An arbitrary pairing function can be denoted as \( f(x,y) \) with pointed brackets \( < x, y > \). If \( < x, y > \) be some pairing function, then the pairing function has to recover each \( x \) and \( y \) arguments from \( < x, y > \). Hence, it is called as two function projection and it can be written as \( \pi_1(x) = x \) and \( \pi_2(x) = y \). If \( z < x, y > \) then, \( \pi_1(x) = x \) and \( \pi_2(x) = y \)

A. Cantor Pairing Function

Given any set \( B \), a pairing function for \( B \) is a 1-1 correspondence from the set of ordered pairs \( B^2 \) to the set \( B \). The set \( B \) is said to be finite with pairing functions, it has fewer than two elements. A pairing function for \( B \) necessarily exists, if \( B \) is infinite. The Cantor’s pairing function [12, 13] for the integers is of the form

\[ c(x, y) = z = \frac{1}{2}(x^2 + 2xy + y^2 - x - 3y + 2) \] (2)

It maps each pair \( (x, y) \) of positive integers to a single positive integer \( c(x, y) \) or \( z \) exists in the literature. There are several variant of Cantor’s exist. They are viz., (i) given any pairing function \( f(x,y) \) for the positive integers, then the function \( f(x + 1, y + 1) - 1 \) is also a Cantor’s pairing function for the non-negative integers. It is defined as

\[ z = f(x + 1, y + 1) - 1 = \frac{1}{2}(x^2 + 2xy + y^2 + 3x + y) \] (3)

(ii) Given any pairing function \( f(x,y) \) for a set \( B \), the Cantor’s pairing function is also obtained by exchanging \( x \) and \( y \) in \( c(x,y) \).

Hence,

\[ c(x, y) = z = \frac{1}{2}(x + y)(x + y + 1) + y \] (4)

The inverse function of the Cantor’s pairing function \( c(x, y) \) is defined as

\[ c^{-1}(z) = \left( \frac{-w(w+1)}{2}, \frac{w(w+3)}{2} \right) - z \] (5)

where \( w = \frac{-1+\sqrt{1+8z}}{2} \)

Cantor Pairing Function- An Example

Consider the two numbers \( x = 47 \) and \( y = 32 \). Then using eqn. (2), we have

\[ c(x, y) = z = \frac{1}{2}(47^2 + 2 \times 47 \times 32) + (3 \times 47) + 32 \] = 3207. To recover the original numbers, inverse of Cantor function shown in eqn. (5) is used. \( w = \frac{1+8(3207)-1}{2} = 79 \) and \( c^{-1}(z) = (3207 \times \frac{79(79+1)}{2}, \frac{79(79+3)}{2} - 3207) \) = (3207 - 3160,3329 - 3207) = (47,32).

B. Rosenberg-Strong Pairing Function

The Rosenberg-Strong pairing function [14] for the non-negative integers is defined by the formula

\[ r(x,y) = (\max(x,y))^2 + \max(x,y) + x - y \] (6)

In the context of the Rosenberg-Strong pairing function, the quantity \( \max(x,y) \) is said to be the shell number of the point \( (x,y) \). The inverse of the Rosenberg-Strong pairing function \( r(x,y) \) is given by the formula,

\[ r^{-1}(z) = (m, m^2 + 2m - z) \text{, otherwise } m = \lfloor \sqrt{z} \rfloor \] (7)

Rosenberg-Strong Pairing Function- An Example

Again consider the two numbers \( x = 47, y = 32 \), using eqn. (6), we have \( r(x,y) = z = (47)^2 + 47 + 47 - 32 = 2271 \).

To recover the original numbers we have \( m = \lfloor \sqrt{z} \rfloor = 47 \).

Using condition 2 in eqn. (7) we get

\[ r^{-1}(z) = 47, (47)^2 + (2 \times 47) - 2271 = (47,32) \]

C. Elegant Pairing Function

If \( x \) and \( y \) are non-negative integers of elegant pairing function. Then \( E(x,y) \) outputs a single non-negative integer that is uniquely associated with that pair [15].

\[ E(x,y) = z = \begin{cases} y^2 + x & x \neq \max(x,y) \\ x^2 + x + y & x = \max(x,y) \end{cases} \] (8)

The inverse function \( E^{-1}(z) \) outputs the pair associated with each non-negative integer \( z \) is defined as

\[ E^{-1}(z) = \begin{cases} \{z - \lfloor \sqrt{z} \rfloor^2, \lfloor \sqrt{z} \rfloor \}, & z - \lfloor \sqrt{z} \rfloor^2 < \lfloor \sqrt{z} \rfloor \\ \{\lfloor \sqrt{z} \rfloor, z - \lfloor \sqrt{z} \rfloor^2 - \lfloor \sqrt{z} \rfloor \}, & z - \lfloor z \rfloor^2 \geq \lfloor \sqrt{z} \rfloor \end{cases} \] (9)
Elegant Pairing Function – An Example

Consider the two numbers $x = 47$ and $y = 32$. Since, $x = \max(x, y)$ then, using eqn. (8), we have $E(x, y) = z = x^2 + x + y = 47^2 + 47 + 32 = 2288$. To recover the original numbers, inverse of Elegant pairing function defined in eqn. (9), condition 2 is used as $E^{-1}(z) = (47, 2288) = (47, 32).$

D. ElGamal Encryption

The ElGamal encryption [16] has its security on the basis of difficulty of calculating DL in a finite field. It consists of three steps. They are viz., key generation, performing encryption using public-key, performing decryption using private-key. To generate a key pair, i.e., private and public-key, first choose a prime $p$ and two random numbers $g$ and $x$, where both $g$ and $x$ are less than $p$. Then, compute

$$y = g^x \mod p$$  \hspace{1cm} (10)

The public key for encryption process is $y, g, p$ and $x$ is the private key. To perform encryption, choose a random $k$, such that $gcd(k, p - 1) = 1$. Then, encryption is performed by computing

$$a = g^k \mod p$$  \hspace{1cm} (11)

The proposed methodology is shown in fig. 1 and 2. It is noted that the ciphertexts are not transmitted to the receiver, instead pairing is performed and the paired numbers are transmitted to the receiver.

IV. PROPOSED METHODOLOGY

For the given $M$, first find the length of the $M$ denoted as $l(M) = l$. Suppose, $M$ is an odd number, append either 0 or a dummy character in $M$. In order to perform pairing, for each $m_i \in M, ASCI1(m_i), i = 1, 2, ..., l$ is found. Then, the adjacent $m_i$ are paired as $f_i(m_{i-1}, m_{i+1}), i = 1, 2, ..., l$ and $k = 1, 3, 5, ... , l - 1$. Pairing of $m_i$ is performed in two ways using said three pairing methods. viz., (i) for two adjacent $m_i$, $i = 1, 3, ..., k$, $ASCI1(m_i)$ values are paired and encryption is performed using ElGamal encryption (ii) encryption is performed using ElGamal first and the resultant $c$ is paired using pairing methods. The proposed methodology is shown in fig. 1 and 2.
Proposed Methodology- An Example

In order to understand the relevance of the work, let M be taken as "KANNAN BABA". Now, \( m_1 = "K", m_2= "A", m_3 = "N", ..., m_{11} = "A" \) i.e., \( m_i \in \mathbb{M} \) and \( c_i \in \mathbb{C} \), where \( i=1,2,3,...,11 \). Then, ASCII\((m_i) = 75\). For ElGamal encryption, let \( p = 10003; \ \text{ASCII}(m_1) = 75; \ g = 34; x = 13 \) and \( k = 25 \). Using eqn. (10), \( y \) is computed as \( y = 34^{25} \mod 10003 = 51195 \). Using eqns. (11) and (12), \( m_i \) is encrypted as \( a = 34^{25} \mod 10003 = 57337 \) and \( b = 51195^{25} \times 75 \mod 10003 = 68229 \). Thus, \( c_i(a,b) = c_i(57337,68229) \). It is noted that the \( c_i(a,b) \) is not transmitted directly to the receiver. Instead, it is converted into single integer using the pairing functions. Let the \( a(b) \) is considered as \( c(x,y) \).

A. ElGamal Encryption Before Pairing Function

Based on the ElGamal encryption, the ciphertext for \( m_1 \) of \( M \) is \( c_1(57337,68227) \).

(i) Cantor Pairing

Cantor pairing is performed using eqn. (2). Let \( c_1(57337,68227) = c(57337,68227) = z = \frac{1}{2}(57337 + 2 \times 57337 \times 68229) + 68229^2 - 57337 - 3 \times 68229 + 2 = 7883279167 \). This is called \( IP_1 \) for calculating \( c^{-1}(z) \) first, \( w \) is calculated using eqn. (5) as \( w = \left[ \frac{\text{7883279167}}{2} \right] = 125564 \). After calculating \( w \), unpair of \( c(x,y) \) is performed using eqn. (5). Now, \( c^{-1}(z) = 7883279167 - 125564(125564+1) = 7883279167 \mod 10003 = 68229 \times 76912^{-1} \mod 10003 = 68229 \times 76439 \mod 10003 = 75 \).

(ii) Rosenberg Strong Pairing

Let \( c \) from the ElGamal encryption is taken as \( r(x,y) = (57337,68229) \). Then, using eqn. (6), \( r(x,y) = (\text{max}(57337,68229) + \text{max}(57337,68229) + 57337 - 68229) = 68229 + 68229 + 57337 - 68229 = 4655253670 \). For unpairing the paired \( r(x,y) \), it must satisfy the condition using eqn. (7). Then, calculate \( m = 68229 \). Here, for unpairing \( r(x,y) \) the two conditions are applied and condition 1 is satisfied so, calculate using eqn. (7) \( r(x,y)^{-1} = (4655253670 - 68229, 68229) \).

(iii) Elegant Pairing

Let \( c \) from the ElGamal encryption is taken as \( E(x,y) = (57337,68229) \). Using eqn. (8), then \( E(x,y) = 68229^2 + 57337 \times 4655253778 \). Now, \( z = IP_1 = 4655253778 \) and for unpairing \( E(x,y) \) using eqn. (9), \( E^{-1}(x,y) = (4655253778 - \sqrt{4655253778^2 - 4655253778}) \) = (57337,68229). Decryption is performed using eqn. (14), \( M = 57337 \mod 10003 = 68229 \times 76912^{-1} \mod 10003 = 68229 \times 76439 \mod 10003 = 75 \).

B. ElGamal Encryption after Pairing Function

Before performing pairing, consider \( x = \text{ASCII}(m_1) = 75 \) and \( y = \text{ASCII}(m_2) = 65 \).
Table I: Encryption of Plaintext before Pairing

| m₀ | ASCII (m₀) | Encryption of Plaintext before Pairing |
|----|------------|---------------------------------------|
|    |            | ElGamal | Cantor | Rosenberg | Elegant |
|    |            | a     | b     | No. of bits | z     | No. of bits | z     | No. of bits | z     | No. of bits |
| K  | 75         | 57337 | 68229 | 16+17=33   | 7.8E+09 | 33         | 4.655E+09 | 33         | 4.66E+09   | 33     |
| A  | 65         | 57337 | 99133 | 16+17=33   | 1.22E+10| 34         | 9.827E+09 | 34         | 9.83E+09   | 34     |
| N  | 78         | 57337 | 98959 | 16+17=33   | 1.22E+10| 34         | 9.792941018| 33         | 9.79E+09   | 34     |
| N  | 78         | 57337 | 98959 | 16+17=33   | 1.22E+10| 34         | 9.792941018| 33         | 9.79E+09   | 34     |
| A  | 65         | 57337 | 99133 | 16+17=33   | 1.22E+10| 34         | 9.827E+09 | 34         | 9.83E+09   | 34     |
| N  | 78         | 57337 | 98959 | 16+17=33   | 1.22E+10| 34         | 9.792941018| 33         | 9.79E+09   | 34     |
| BS | 32         | 57337 | 44423 | 16+16=32   | 5.18E+09 | 33         | 3287601820 | 32         | 3.9E+09    | 32     |
| B  | 66         | 57337 | 76042 | 16+17=33   | 8.9E+09 | 34         | 5782443101 | 33         | 5.7E+09    | 33     |
| A  | 65         | 57337 | 99133 | 16+17=33   | 1.22E+10| 34         | 9.827E+09 | 34         | 9.83E+09   | 34     |
| B  | 66         | 57337 | 76042 | 16+17=33   | 8.9E+09 | 34         | 5782443101 | 33         | 5.7E+09    | 33     |
| A  | 65         | 57337 | 99133 | 16+17=33   | 1.22E+10| 34         | 9.827E+09 | 34         | 9.83E+09   | 34     |
| Total No. of bits | 362 | 372 | 366 | 369 |

Table II: Encryption of Plaintext after Pairing

| m₀ | ASCII (m₀) | Encryption of Plaintext after Pairing |
|----|------------|---------------------------------------|
|    |            | Cantor with ElGamal | Rosenberg with ElGamal | Elegant with ElGamal |
|    |            | z     | a     | b     | No. of Bits | z     | a     | b     | No. of Bits | z     | a     | b     | No. of Bits |
| K  | 75         | 9945  | 57337 | 66896 | 16+17=33   | 5710  | 57337 | 54347 | 16+16=32   | 5765  | 57337 | 84381 | 16+17=33   |
| A  | 65         | 12324 | 57337 | 35054 | 16+16=32   | 6162  | 57337 | 17527 | 16+15=31   | 6240  | 57337 | 16483 | 16+15=31   |
| N  | 78         | 10361 | 57337 | 61328 | 16+16=32   | 6149  | 57337 | 17701 | 16+15=31   | 6149  | 57337 | 17701 | 16+15=31   |
| B  | 32         | 4883  | 57337 | 50031 | 16+16=32   | 4388  | 57337 | 79734 | 16+17=33   | 4388  | 57337 | 79734 | 16+17=33   |
| A  | 65         | 8711  | 57337 | 60335 | 16+16=32   | 4421  | 57337 | 17752 | 16+15=31   | 4421  | 57337 | 17752 | 16+15=31   |
| B  | 66         | 2210  | 57337 | 70423 | 16+17=33   | 4355  | 57337 | 41713 | 16+16=32   | 65    | 57337 | 99133 | 16+17=33   |
| Total No. of Bits | 194 | 190 | 192 | BS-Blank Space |

V. RESULTS AND DISCUSSION

The proposed methodology is implemented in VC++ with IDE 5.0. The said three pairing functions are applied in the M before and after performing encryption using ElGamal cryptosystem with different file size. The security level is measured using All Block Cipher (ABC) Universal Hackman tool which uses dictionary attack. The encryption time, decryption time and the security level are recorded in table 3, 4 and 5 respectively. It is evident from tables III and IV that the time taken for performing encryption and decryption is increasing before pairing than after pairing.

In before pairing, for each m₀ two different ciphertexts are produced when ElGamal encryption is applied and the magnitude of c₀ depends on the modulus p and its value ranging from 0 to p – 1. Suppose, if M has n number of m₀, then 2n number of ciphertexts are produced. While applying pairing, it is reduced to again n. The number of bits involved in pairing is increasing for all pairing schemes when it is compared with conventional ElGamal. It is observed from the table I that even though the number of bits producing the ciphertext before pairing is larger than the conventional ElGamal encryption, the security is increasing before pairing because, the ciphertexts obtained from the ElGamal encryption is converted into single ciphertext for each character and it is not able to predictable by the adversary.

In the case of after pairing, ASCII(m₀) and ASCII(m₀ + 1) are combined which is reduced to one z₀. It is evident from table II, that the number of bits required for ciphertexts are reduced approximately by half of the bits required for the plaintext when conventional and before ElGamal encryption which results in decreasing the encryption and decryption time. Further, it is noticed from table 2 that the number of ciphertexts are exactly equal to the number of characters in M contrary to the 2c₀ produced by the conventional ElGamal. It also enhances the security because original mᵢM are not used for encryption but it uses only the paired value.

Table III: Encryption Time Produced by ElGamal Encryption Before and After Pairing

| File Size (in mb) | Existing Method (in ms) | Proposed Method (in ms) |
|------------------|------------------------|------------------------|
|                  | Before Pairing         | After Pairing          |
|                  | ElG                  | CPEL          | EPEL          | ELCP              | ELRP         | ELIP          |
| 1                | 2033                | 3703          | 3600          | 3241              | 3316         | 3346          | 3315         |
| 2                | 4710                | 6463          | 5897          | 5825              | 6212         | 5835          | 5676         |
| 3                | 7302                | 9892          | 9051          | 8763              | 9735         | 9125          | 8803         |
| 4                | 9769                | 13374         | 12260         | 11949             | 13110        | 12248         | 11861        |
| 5                | 12323               | 16751         | 15547         | 14978             | 16623        | 15484         | 14978        |
| Avg.             | 7347.4              | 10037         | 9225          | 8951.2            | 9879.2       | 9207.6        | 8926.6       |
Enhancing the Security in ElGamal Cryptosystem using Paring Functions

Table IV: Decryption Time Produced by ElGamal Encryption Before and After Paring

| File Size (in mb) | Existing Method | Proposed Method | Before Paring (in ms) | After Paring (in ms) |
|------------------|-----------------|-----------------|---------------------|---------------------|
|                  | ElG             | CPEL            | RPEL                | EPEL                | ELCP               | ELRP               | ELEP               |
| 1                | 2741            | 3736            | 2344                | 3316                | 3706               | 3295               | 3203               |
| 2                | 4643            | 6507            | 5931                | 5769                | 6394               | 5973               | 5724               |
| 3                | 7172            | 9922            | 9114                | 8813                | 9784               | 9133               | 8803               |
| 4                | 9945            | 13408           | 12384               | 11987               | 13264              | 12271              | 11812              |
| 5                | 12209           | 16509           | 15573               | 15098               | 16766              | 15605              | 15137              |
| Avg              | 7342            | 10016.4         | 9269.2              | 8996.6              | 9922               | 94.4               | 8935.8             |

Fig. 3: Encryption Time Produced by ElGamal Encryption Before and After Paring

Fig. 4: Decryption Time Produced by ElGamal Encryption Before and After Paring

Table V: Security Level Produced by ElGamal Encryption Before and After Paring

| File Size (in mb) | Existing Method | Proposed Method | Before Paring (in %) | After Paring (in %) |
|------------------|-----------------|-----------------|---------------------|---------------------|
|                  | ElG             | CPEL            | RPEL                | EPEL                | ELCP               | ELRP               | ELEP               |
| 1                | 89              | 94              | 93                  | 93                  | 94                 | 94                 | 93                 |
| 2                | 86              | 93              | 94                  | 94                  | 94                 | 95                 | 93                 |
| 3                | 85              | 94              | 94                  | 94                  | 94                 | 94                 | 93                 |
| 4                | 85              | 94              | 93                  | 93                  | 95                 | 94                 | 92                 |
| 5                | 87              | 94              | 94                  | 94                  | 94                 | 93                 | 93                 |
| Avg              | 86.4            | 93.8            | 93.2                | 93.6                | 94.4               | 94.2               | 94                 |

Fig. 5: Security Level Produced by ElGamal Encryption Before and After Paring

VI. CONCLUSION

An enhanced version of ElGamal encryption with three different pairing methods is thought of and implemented them successfully. The experimental results clearly indicated that there is an increase in the security level of ElGamal encryption when pairing function are used in it. It is noted that the security level is increasing in ElGamal by 94.4%, 94.2% and 94% after pairing than before pairing by 93.8%, 93.2% and 93.6% but the conventional ElGamal produces only 86.4% security level. It is observed from the experimental results that that the number of bits has been reduced when pairing the plaintext is performed after encryption rather than before pairing. Further, it is evident from the tables that the security is increasing when the encryption is performed after pairing the plaintext than before pairing for all the three pairing methods taken. Among the three pairing methods, Cantor pairing provides more security than other two counterparts when encryption is performed before and after pairing. Also, Rosenberg’s Strong pairing scheme is the best because it reduces the number of bits with respect to all pairing which in turn increases the transmission speed of ciphertext when it is transmitted through the communication channel.

REFERENCES

1. Andreas V. Meier, “The ElGamal Cryptosystem”, In the proceedings of the CRYPTO '84, pp 1-13, 2005.
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, 1996.
3. Arnold L. Rosenberg, “Efficient Pairing Functions—and Why You Should Care”, In International Journal of Foundations of Computer Science, Vol. 14, pp. 1-7, 2003.
4. Reddiah, “A Study on. Pairing Functions for Cryptography”, International Journal of Computer Applications, pp. 0975 – 8887, 2016.
5. B Hari Krishna, I Raja Sekhar Reddy, Kiran Sk and Pradeep Kumar Reddy, “Multiple text encryption, Key entrenched, distributed cipher using pairing functions and transposition ciphers”, Wireless Communications, Signal Processing and Networking, 2016.
6. Myungsun Kim, Jihye Kim and Jung Hee Cheon, “Compress Multiple Ciphertexts Using Elgamal Encryption Schemes”, pp. 1-12, 2010, https://eprint.iacr.org/2012/243.pdf.
7. Steven D. Galbraith1, Kenneth G. Paterson1, and Nigel P. Smart, “Pairings for Cryptographers”, Discrete Applied Mathematics, Vol. 156, pp. 3113–3121, 2008.
8. Mehmet Sabur Kiraz and Osmanbey Uzunkol, “Still Wrong Use of Pairings in Cryptography”, Applied Mathematics and Computation, Elsevier, Vol. 333, pp. 467-479, 2018.
9. Dan Boneh and Matthew Franklin, “Identity-Based Encryption from the Weil Pairing”, Lecture Notes in Computer Science, Springer-Verlag, pp. 213–229, 2001.
10. Neal Koblitz and Alfred Menezes, “Pairing-Based Cryptography at High Security Levels”, IMA International Conference on Cryptography and Coding, Springer, pp 13-36, 2005.
11. Pairing function available at: https://www.cs.upc.edu/~alvarez/calculabilitat/enumerabilitat.pdf.
12. Nadia El Mrabet, Marc Joy, “Guide to Pairing-Based Cryptography”, 1st Edition, Chapman and Hall/CRC, New York, 2017.
13. Patrick Cegielski and Denis Richard, “Decidability of the Theory of the Natural Integers with the Cantor Pairing Function and the Successor”, Theoretical Computer Science, Elsevier, Vol. 257, pp. 51-77, 2001.
14. M P Szudzik, “The Rosenberg-Strong Pairing Function”, Discrete Mathematics, 2019.
15. Matthew Szudzik, “An Elegant Pairing Function”, Wolfram Research, Inc., 2006.
16. Bruce Schneier, “Applied Cryptography”, John Wiley & Sons, Inc., 1996.

AUTHORS PROFILE

Mani. K., received his MCA and M.Tech. from the Bharathidasan University, Trichy, India in Computer Applications and Advanced Information Technology respectively. Since 1989, he has been with the Department of Computer Science at the Nehru Memorial College (Autonomous), affiliated to Bharathidasan University where he is currently working as an Associate Professor. He finished graduation in Operations Research from Operational Research Society of India, Kolkatta. He completed his PhD in Cryptography with primary emphasis on evolution of framework for enhancing the security and optimizing the run time in cryptographic algorithms. He published and presented around 45 research papers at international journals including Springer, Inderscience and conferences. His research areas are cryptology and Coding theory.

Barakath Begam. A., received her M.Sc. and M.Phil. From Bharathidasan University, Trichy, India. Currently, she is pursuing her Ph.D. in Cryptography, Bharathidasan University, Trichy. Her research interest is on Cryptography, Network Security.