Computer Science and Engineering 2017, 7(2): 29-44
DOI: 10.5923/j.computer.20170702.01

Data and Information Security in Modern World by using Elliptic Curve Cryptography

Obaidur Rahaman

European University of Bangladesh, Department of Computer Science and Engineering, Bangladesh

Abstract Data and Information Security has become very important in today's modern world, as a result of these various methods are adopted to bypass it. With the advent of the internet, security has became a major concern and the history of security allows a better understanding of the emergence of security technology. The internet structure itself allowed for many security threats to occur. The mechanism of the internet, when modified can reduce the possible attacks that can be sent across the network. By knowing the attack methods, allows for the appropriate security to emerge. Many modern spare secure themselves from the internet by means of Elliptic Curve Cryptography, Text Encryption and Decryption process and Card Shuffling Process. It provides higher level of security with lesser key size compared to other Cryptographic techniques. A new technique has been proposed in this paper where the classic technique of mapping the characters to affine points in the elliptic curve has been removed. The corresponding ASCII values of the plain text are paired up. The paired values serve as input for the Elliptic curve cryptography. This new technique has been avoided for the costly operation of mapping and the need to share the common lookup table between the sender and the receiver. The algorithm is designed in such a way that it can be used to encrypt or decrypt any type of script with defined ASCII values.

Keywords Cryptography, Elliptic curve cryptography, Encryption and Decryption Process, Card Shuffling Process

1. Introduction

Data and Information security is one of the most important issues in the modern civilization. In this modern world most of the data are transferred and stored by using internet. So it is important to secure our data from unauthorized access. Computer scientists are developing different types of mechanism to secure data. As old mechanisms are destroyed by different types of unauthorized attacks, computer scientists are developing new and modern types of security mechanism to protect data. Data are encrypted thus unauthorized user can't get actual data and decrypted to use by authorized user. Cryptology is the study of secure communications, which encompasses both cryptography and cryptanalysis [1]. A strong data encryption and decryption technique is required to provide confidentiality of sensitive data from security attacks. The idea of information security leads to the evolution of Cryptography. In other words, Cryptography is the science of keeping information secure. It involves encryption and decryption of messages. There have been many known cryptographic algorithms. The crux of any cryptographic algorithm is the "seed" or the "key" used for encrypting/decrypting the information. Many of the cryptographic algorithms are available publicly, though some organizations believe in having the algorithm a secret. The general method is in using a publicly known algorithm while maintaining the key secret. We have studied application of Elliptic Curves over card shuffling logic for traditional key exchange and encryption of text. We have implemented both and proposed a scheme for encryption of images. It was partially accomplished for a small size image.

* Corresponding author:
obaidur.rahman988@gmail.com (Obaidur Rahaman)
Published online at http://journal.sapub.org/computer

2. Implementation of Elliptic Curve Cryptography on Text and Image

Describe a brief background of encryption/decryption and key exchange are using in ECC thesis paper "Implementation of Elliptic Curve Cryptography (ECC) on Text and Image" [7]. Used C++ to implement text encryption and mapping table to map the ASCII value of Elliptic curve coordinate. Reverse mapping is used while decryption.

Global Public Elements

$E_q(a,b)$ elliptic curve with parameters a, b & q in the equation

$Y^2 \bmod q=(X^3+aX+b) \bmod q$

Q Base point on elliptic curve

User A Key Generation

Select private key k_A, $k_A < n$

Calculate public P, $P=k_A \times Q$

User B Key Generation

Select private key k_B, $k_B<n$

Calculate public M, $M=k_B \times Q$

Generation of Secret Key by user A

$P_1=K=k_A \times M$

Generation of Secret Key by user B

$P_2=K=k_B \times P$

The two calculations produce the same result because

$k_A \times M = k_A \times (k_B \times Q) = k_B \times (k_A \times Q) = k_B \times P$

To break this scheme, an attacker would be needed to be able to compute k given G & kG, which is found to be tough.

For example, scalar multiple k is 5; G ≡ (2,2) then let 5G= D≡(153,108) for a=0, b=-4, q=211. Given the values of G and D, it is difficult to find out the scalar multiple k=5.

2.1. Elliptic Curve Encryption and Decryption

1. Consider a message 'Pm' sent from A to B. 'A' chooses a random positive integer 'k', a private key 'n' and generates the public key P = n × G and produces the ciphertext 'Cm' consisting of pair of points **$Cm=\{kG, Pm + kP_B\}$**

where G is the base point selected on the Elliptic Curve, $P_B= n_B \times G$ is the public key of B with private key 'n_B'

2. To decrypt the ciphertext, B multiplies the 1st point in the pair by B's secret & subtracts the result from the 2nd point **$Pm + kP_B - n_B(kG) = Pm + k(n_B G) – n_B(kG)=Pm$**

2.2. Software Implementation

We have implemented the key exchange and the text encryption procedure using C++.

a. Defining basic functions that are to be used in all programs:-

Various codes under this section are:

1. Code to find the multiplicative inverse of an integer for a given prime number:-

For this code, we have used the extended Euclid Algorithm whereby the intermediate terms are less than the prime numbers. This prevents the intermediate terms from exceeding the corresponding prime number.

For example: Consider the prime number 23

Now $6^{-1} \bmod 23=4$,

Since, $(6\times4) \bmod 23=1$

2. Code to generate the points on an Elliptic curve:-

As there is constant need for a database of the elliptic curve points, a code to scan all Y co-ordinates that satisfy the elliptic curve equation for the given X co-ordinate has been included.

Equation of the elliptic curve: $y^2 \bmod p=(x^3+ax+b) \bmod p$

Where, p is a prime number.

Algorithm: Inputs: p, a, b

a. Enter the input data.

b. x=[0: p-1]

c. For each value of x, check which values of y from 0 to (p-1) satisfies the equation.

d. Display the required point.

For example: p=211, a=0, b=-4

Given Table (1): Elliptic curve points

Table (1). Elliptic curve points

X	Y
167	30
167	181
179	12
191	15

3. Code to find the public key:-

Input: X_G=X co-ordinate of G

Y_G=Y co-ordinate of G

n_A be the private key. (A scalar multiple)

Let P_A be the public key. $P_A= n_A \times G$

We carry out recursive addition of point G for n_A times to get the point P_A.

For example: G= (2, 2), n_A=5, (153,108) = 5(2, 2). So, P_A= (153,108) is the public key for private key 5.

4. Code for encoding and decoding:-

a. Mapping-1 is used to convert an integer into a corresponding elliptic curve points from our database.

b. Mapping-2 is used to convert an elliptic curve point into its corresponding integer from our database.

c. Reverse Mapping-1 does the same function as Mapping-2.

d. Reverse Mapping-2 does same function as Mapping-1.

These codes perform scanning operations shown below on Table (2).

For example:

Table (2). Codes Scanning Operations

Point No.	X	Y
15	16	100
16	16	101
17	17	30
18	17	181
19	19	37
20	19	174
21	20	20
22	20	191
23	21	87
24	21	124

b. Program for secret key exchange:-

The algorithm for secret key exchange is the same as in Section IV. The recursive addition used while finding the order is used to find the intermediate scalar multiple by restricting the addition at the required scalar multiplication.

c. Program for text encryption:-

The text encryption procedure has used the in-built feature of C++ to assign the ASCII value of a character to an integer variable when the latter is equated to the former.

2.2.1. Text Encryption Procedure

Requirement: The order of the curve (i.e. the number of points on the curve) should be greater than 126.

Constraint: There should be no complete blank line in the text.

1. Read a character from the plain text i.e. 'a'.
2. Gets its ASCII value into an integer variable (say I=97).
3. Select the point {say E(17,30)} on the Elliptic curve corresponding to the integer I=97 as per our database. (Mapping-1).
4. Now this point is encrypted as per Section V.
5. Let the encrypted point be E´≡ (21,124).
6. Now this point E´ is mapped once again to the database to obtain new integer. (Mapping-2).
7. The new integer corresponding to E´ is I´(say 43).
8. This new integer I' is converted to data consisting of two parameters
   a. principle ASCII character ($) which act as an index
   b. Page no. (say N=1) to which the corresponding index belongs to.
9. These two parameters are sent into two different files which are transmitted.

2.2.2. Text Decryption Procedure

1. The encrypted character ($) and the corresponding page no. (N) are read from the received files.
2. These two parameters are used to calculate back the integer ***I´*.**
3. Then reverse mapping-2 is carried out to convert the integer ***I´*** to a point ***E´*** from the database.
4. The point ***E´*** is decrypted as per Section V to get a point E.
5. Now reverse mapping-1 is carried out on point E to get the integer I from our database.
6. The ASCII character with ASCII value I is the plain text character which was originally encrypted.

2.2.3. Features of Encryption Procedure

Generally, the encrypted curve points are transmitted on line. But in English language, certain letters have certain fixed probability of usage. So for a particular letter, if a certain elliptic curve point is transmitted always, the attacker can decrypt the elliptic curve point by checking its frequency of usage. In short, this technique has a flaw of simple substitution ciphering technique.

This problem is solved by using many-to-one mapping for characters and the original characters are differentiated by using their corresponding page numbers. So, different characters may be encrypted to the same character.

Solution for conversion from non-printable characters to printable character after encryption is as follows:

It has been observed that the ASCII characters from 0 to 31 are non-printable. So if the encrypted character is found to be within this range, there is additional calculation which is carried out.

In such a case, a tilde (~) is transmitted and the ASCII value of the encrypted character is incremented by 32 for transmission as a printable character.

On the decryption side, a reverse calculation is done when a tilde (~) is detected.

Result of text encryption p-decryption program:

Plain text

This is document number 2 for encryption.

Encrypted text

G09I09I0>Eop#I^~60^p#~>IR0k0]ER0I^oR~7+~69E^D EEE~+G

Decrypted text

This is document number 2 for encryption.

2.2.4. Image Encryption Procedure

The image encryption procedure is based on encrypting the intensity and thus converting it into a new intensity. This new intensity is decrypted at the receiver side to obtain the original intensity.

Steps are:

1. Read the intensity I from the image intensity matrix.
2. Convert the intensity into an elliptic curve point E using Mapping-1.
3. Encrypt the Elliptic curve point to a new point(E').
4. Using Mapping-2, the new point is converted to a corresponding integer M.
5. This integer M is used to calculate the new encrypted intensity I´ and page number P.

2.2.5. Image Decryption Procedure

1. The encrypted intensity I' and the page number (P) are read from the received files.
2. These 2 parameters are used to calculate the integer M.
3. Using reverse mapping-2, the integer M is converted to encrypted elliptic curve point E'.
4. The point E' is decrypted to get the original point E.
5. By reverse mapping-1, the original intensity I is obtained.

2.2.6. Features of Image Encryption Procedure

1. The maximum image size is 32×32 due to restriction on the number of elements in the global array.
2. Any image of size greater than 32×32 needs to be cropped and sent as multiple files.
3. Any image or part of it smaller than 32×32 needs to be brought to 32×32 size by padding zeros in the required locations.

The procedure was implemented on Matlab images and the images were successfully encrypted and decrypted with zero error.

3. Algorithm for Text Encryption and Decryption over Elliptic Curve using Card Shuffling Logic

Many researchers implemented text encryption and decryption over elliptic curve using ASCII values or various types of mapping on common look up table. In this research work we want to introduce a playing card shuffling logic for text encryption and decryption. Card shuffling logic is used to make a dynamic and secure mapping table. We shuffle our playing cards using three randomly selected points over generated elliptic curve and use ElGamal Cryptosystem to encrypt and decrypt these points.

All the procedure of the proposed algorithm is given below. Figure (1a) shows our proposed mechanism.

3.1. Key Generation for Text Encryption and Decryption

The key generation is same as ElGamal Cryptosystem using Elliptic Curve.

Receiver chooses $E_p(a,b)$ with an elliptic curve over GF(p), e_1 is a point on elliptic curve, $e_1=(x_1, y_1)$. $e_2=d\times e_1$, where d is the private key.

Receiver announces public keys $E_p(a,b)$, e_1 and e_2.

3.2. Encryption Process

Sender receives public keys $E_p(a,b)$, e_1 and e_2 from receiver. Sender chooses three points over elliptic curve P_1, P_2 and P_3. Sender uses those points to shuffle 52 cards.

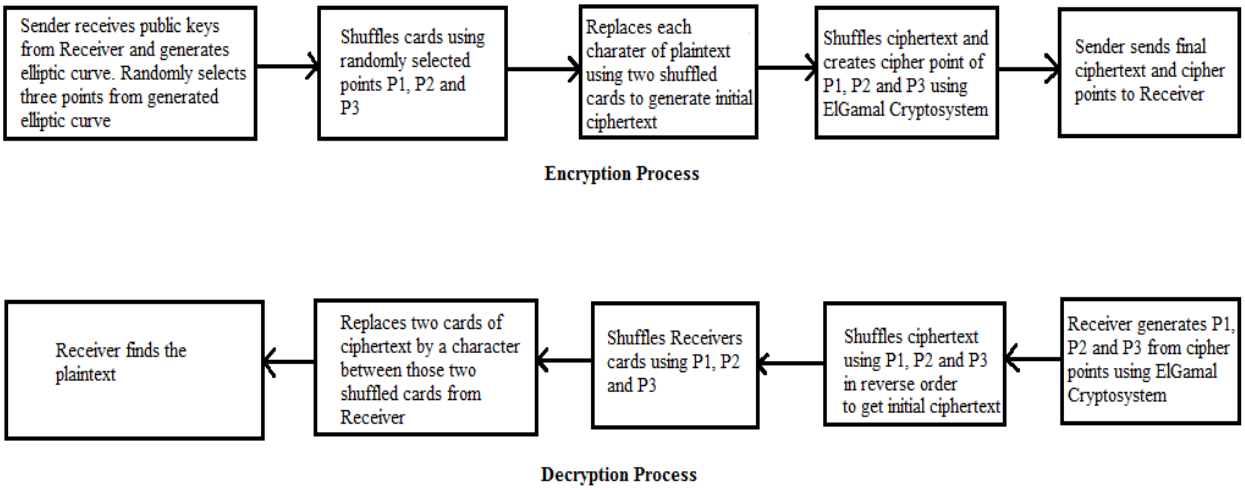


Figure (1a). Proposed Mechanism

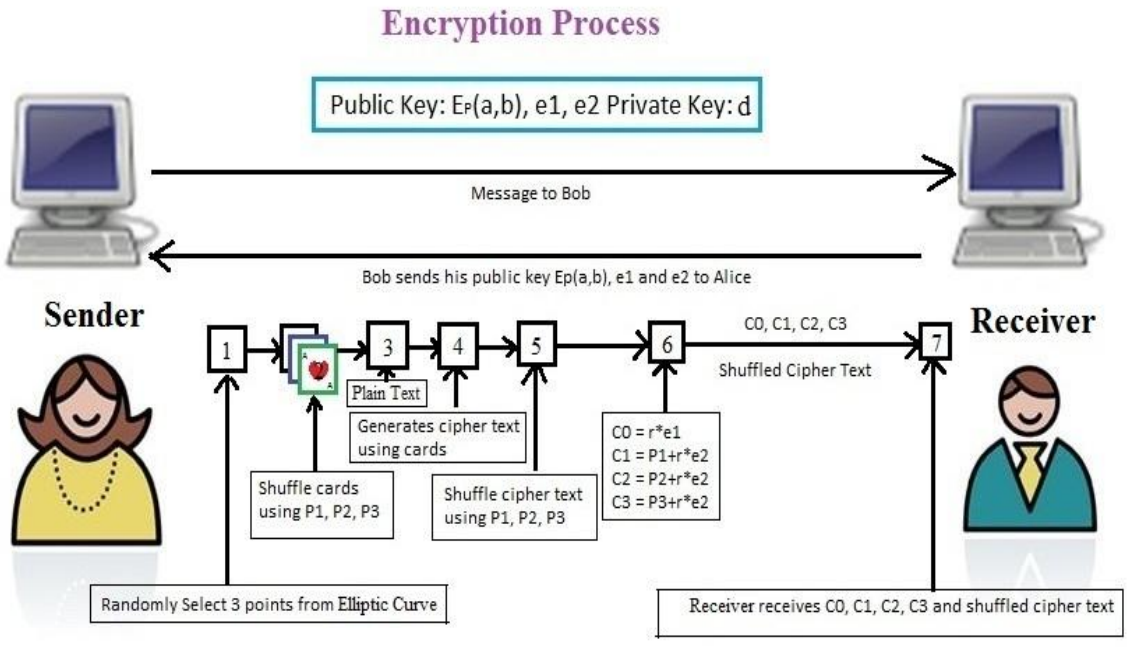


Figure (1b). Encryption process of proposed method

3.2.1. Card Shuffling Process

Consider 4×13 matrix for cards,

Here 2H= 2 of Hearts, 3H= 3 of Hearts…… 9H=9 of Hearts, #H=10 of Hearts (10H has three characters but in our process need 2 character to present each card; so, we replace 10 by special character #) JH= Jack of Hearts, QH=Queen of Hearts, KH=King of Hearts and AH=Ace of Hearts.

2C= 2 of Clubs, 3C= 3 of Clubs…… 9C=9 of Clubs, #C=10 of Clubs (10C has three characters but in our process need 2 character to present each card; so, we replace 10 by special character #) JC= Jack of Clubs, QC=Queen of Clubs, KC=King of Clubs and AC=Ace of Clubs.

2D= 2 of Dice, 3D= 3 of Dice…… 9D=9 of Dice #D=10 of Dice (10D has three characters but in our process need 2 character to present each card; so, we replace 10 by special character #) JD= Jack of Dice, QD=Queen of Dice, KD=King of Dice and AD=Ace of Dice.

2S= 2 of Spade, 3S= 3 of Spade…… 9S=9 of Spade #S=10 of Spade (10S has three characters but in our process need 2 character to present each card; so, we replace 10 by special character #) JS= Jack of Spade, QS=Queen of Spade, KS=King of Spade and AS=Ace of Spade.

Position of Matrix (1)

Matrix (1). Card Shuffling Process

2H	2C	2D	2S
3H	3C	3D	3S
4H	4C	4D	4S
5H	5C	5D	5S
6H	6C	6D	6S
7H	7C	7D	7S
8H	8C	8D	8S
9H	9C	9D	9S
#H	#C	#d	#S
JH	JC	JD	JS
QH	QC	QD	QS
KH	KC	KD	KS
AH	AC	AD	AS

Matrix (2). Position of Card Shuffling Process

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)
(3,0)	(3,1)	(3,2)	(3,3)
(4,0)	(4,1)	(4,2)	(4,3)
(5,0)	(5,1)	(5,2)	(5,3)
(6,0)	(6,1)	(6,2)	(6,3)
(7,0)	(7,1)	(7,2)	(7,3)
(8,0)	(8,1)	(8,2)	(8,3)
(9,0)	(9,1)	(9,2)	(9,3)
(10,0)	(10,1)	(10,2)	(10,3)
(11,0)	(11,1)	(11,2)	(11,3)
(12,0)	(12,1)	(12,2)	(12,3)

For point $P_1(x,y)$: each card x steps right and y step down

If card current position is (i,j) in the matrix. We can find new position (i_1,j_1) by applying $P_1(x,y)$ for m×n matrix

$$i_1 = (i+y) \bmod n \quad (1)$$

$$j_1 = (j+x) \bmod m \quad (2)$$

In this case matrix size is 4×13

If $P_1(x,y)$ =(2,3) from Matrix(1) 3H current position is (1,0)

So, x=2, y=3, i=1 and j=0. Now 3H new position is

$i_1 = (1+3) \bmod 13 = 4$

$j_1 = (0+2) \bmod 4 = 2$

3H new position (4,2). That means 3H moves 2 steps right and 3 steps down

From Matrix (1) AS position is (12,3)

So, x=2, y=3, i=12 and j=3. Now AS new position is

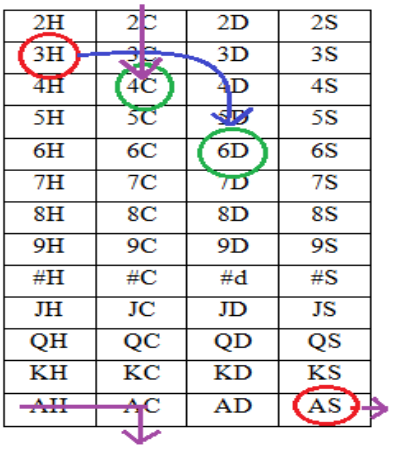
$i_1 = (12+3) \bmod 13 = 2$

$j_1 = (3+2) \bmod 4 = 1$

AS new position (2,1). In this case when there is no scope to go right (at the last position of row) it will continue with current row first position. This is also similar with the column when there is no scope to go down (at the last position of column) it will continue with current column first position.

Two operations $P_1(x,y)$ are shown below on Table (3).

Table (3). Appling two operations of $P_1(x,y)$



2H	2C	2D	2S
3H	3C	3D	3S
4H	4C	4D	4S
5H	5C	5D	5S
6H	6C	6D	6S
7H	7C	7D	7S
8H	8C	8D	8S
9H	9C	9D	9S
#H	#C	#d	#S
JH	JC	JD	JS
QH	QC	QD	QS
KH	KC	KD	KS
AH	AC	AD	AS

Applying P_1 for all the cards we get.

(0,0) => (3,2)	(0,1) => (3,3)	(0,2) => (3,0)	(0,3) => (3,1)
(1,0) => (4,2)	(1,1) => (4,3)	(1,2) => (4,0)	(1,3) => (4,1)
(2,0) => (5,2)	(2,1) => (5,3)	(2,2) => (5,0)	(2,3) => (5,1)
(3,0) => (6,2)	(3,1) => (6,3)	(3,2) => (6,0)	(3,3) => (6,1)
(4,0) => (7,2)	(4,1) => (7,3)	(4,2) => (7,0)	(4,3) => (7,1)
(5,0) => (8,2)	(5,1) => (8,3)	(5,2) => (8,0)	(5,3) => (8,1)
(6,0) => (9,2)	(6,1) => (9,3)	(6,2) => (9,0)	(6,3) => (9,1)
(7,0) => (10,2)	(7,1) => (10,3)	(7,2) => (10,0)	(7,3) => (10,1)
(8,0) => (11,2)	(8,1) => (11,3)	(8,2) => (11,0)	(8,3) => (11,1)
(9,0) => (12,2)	(9,1) => (12,3)	(9,2) => (12,0)	(9,3) => (12,1)
(10,0) => (0,2)	(10,1) => (0,3)	(10,2) => (0,0)	(10,3) => (0,1)
(11,0) => (1,2)	(11,1) => (1,3)	(11,2) => (1,0)	(11,3) => (1,1)
(12,0) => (2,2)	(12,1) => (2,3)	(12,2) => (2,0)	(12,3) => (2,1)

For point $P_2(x,y)$: each card x steps left and y step up
If card current position is (i,j) in the matrix. We can find new position (i_2,j_2) by applying $P_2(x,y)$ for m×n matrix.

$$i_2 = (i-y) \bmod n \quad (3)$$

$$j_2 = (j-x) \bmod m \quad (4)$$

In this case matrix size 4×13

Matrix (3). After Appling $P_1(x,y)$ operations

QD	QS	QH	QC
KD	KS	KH	KC
AD	AS	AH	AC
2D	2S	2H	2C
3D	3S	3H	3C
4D	4S	4H	4C
5D	5S	5H	5C
6D	6S	6H	6C
7D	7S	7H	7C
8D	8S	8H	8C
9D	9S	9H	9C
#D	#S	#H	#C
JD	JS	JH	JC

If $P_2(x,y)$ =(3,5) from **Matrix (3)** after doing $P_1(x,y)$ operation, 3H current position is (4,2)
So, x=3, y=5, i=4 and j=2. Now 3H 2nd new position is
i_2 = (4-5) mod 13 = 12
j_2 = (2-3) mod 4 = 3
3H 2nd new position (12,3). That means 3H moves 3 steps left and 3 steps up
In this case when there is no scope to go left (at the first position of row) it will continue with current row last position. This is also similar with the column when there is no scope to go up (at the first position of column) it will continue with current column last position.
After applying $P_1(x,y)$ operation from Matrix (3) AS current position is (2,1)
So, x=3, y=5, i=2 and j=1. Now AS new position is
i_1 = (2-5) mod 13 = 10
j_1 = (1-3) mod 4 = 2
AS 2nd new position (10,2).
Two operations of $P_2(x,y)$ are shown below.

Table (4). Two operations of $P_2(x,y)$

QD	QS	QH	QC
KD	KS	KH	KC
AD	AS	AH	AC
2D	2S	2H	2C
3D	3S	3H	3C
4D	4S	4H	4C
5D	5S	5H	5C
6D	6S	6H	6C
7D	7S	7H	7C
8D	8S	8H	8C
9D	9S	9H	9C
#D	#S	#H	#C
JD	JS	JH	JC

Applying P_2 for all the cards we get,

(0,0) => (8,1)	(0,1) => (8,2)	(0,2) => (8,3)	(0,3) => (8,0)
(1,0) => (9,1)	(1,1) => (9,2)	(1,2) => (9,3)	(1,3) => (9,0)
(2,0) => (10,1)	(2,1) => (10,2)	(2,2) => (10,3)	(2,3) => (10,0)
(3,0) => (11,1)	(3,1) => (11,2)	(3,2) => (11,3)	(3,3) => (11,0)
(4,0) => (12,1)	(4,1) => (12,2)	(4,2) => (12,3)	(4,3) => (12,0)
(5,0) => (0,1)	(5,1) => (0,2)	(5,2) => (0,3)	(5,3) => (0,0)
(6,0) => (1,1)	(6,1) => (1,2)	(6,2) => (1,3)	(6,3) => (1,0)
(7,0) => (2,1)	(7,1) => (2,2)	(7,2) => (2,3)	(7,3) => (2,0)
(8,0) => (3,1)	(8,1) => (3,2)	(8,2) => (3,3)	(8,3) => (3,0)
(9,0) => (4,1)	(9,1) => (4,2)	(9,2) => (4,3)	(9,3) => (4,0)
(10,0) => (5,1)	(10,1) => (5,2)	(10,2) => (5,3)	(10,3) => (5,0)
(11,0) => (6,1)	(11,1) => (6,2)	(11,2)=> (6,3)	(11,3)=>(6,0)
(12,0) => (7,1)	(12,1) => (7,2)	(12,2)=> (7,3)	(12,3)=> (7,0)

After changing all the position we get the matrix (4).

Matrix (4). Position after operation of $P_2(x,y)$

4C	4D	4S	4H
5C	5D	5S	5H
6C	6D	6S	6H
7C	7D	7S	7H
8C	8D	8S	8H
9C	9D	9S	9H
#C	#D	#S	#H
JC	JD	JS	JH
QC	QD	QS	QH
KC	KD	KS	KH
AC	AD	AS	AH
2C	2D	2S	2H
3C	3D	3S	3H

For $P_3(x,y)$ each card shift x steps, each card shift y steps shown by Table (5).

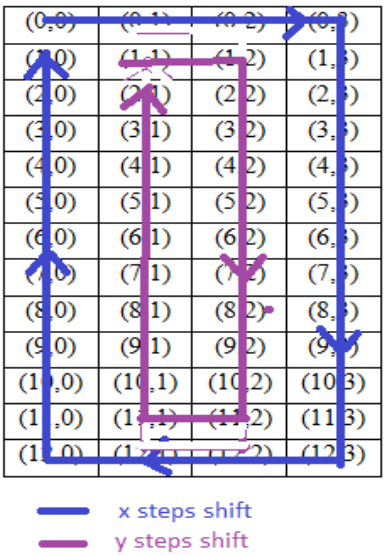
x steps shift contains with those cards which are in the boundary of the matrix. And y steps shift with 2nd and 3rd column except first and last row of the matrix.

$$\text{Number of members of x shift is } x_m = 2(n + 2) \text{ [for m×n matrix]} \quad (5)$$

$$\text{Number of members in y shift is } y_m = 2(n - 2) \text{ [for m×n matrix]} \quad (6)$$

Number of member of x shift is 2(13+2) or 30 and number of member in y shift is 2(13-2) or 22.
Now modifying our selected elliptic curve point is $P_3(x \bmod x_m, y \bmod y_m)$

Table (5). P3(x,y) card shifting



Shifting x steps (Clockwise): Here we representing each card position by (i,j) and all the points which belongs to x shift are added to an array, size of the array is number of member for the x shifts and k is the array position. We have to determine new position k_n for each card after x shifts.

Suppose our randomly selected third point over elliptic curve is (2,1)
Now, $P_3(2 \bmod 30, 1 \bmod 26)$ or $P_3(2,1)$
Now, x = 2
Shifting 2 steps

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
i	0	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	12	12	12	11	10	9	8	7	6	5	4	3	2	1
j	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0

k_n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
i	2	1	0	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	12	12	12	11	10	9	8	7	6	5	4	3
j	0	0	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	2	1	0	0	0	0	0	0	0	0	0	0

Now, move card k=29 to new position k_n=1

move card k=28 to new position k_n=0

… … …

… … …

… … …

move card k=0 to new position k_n=2

New position $k_n=(k+x) \bmod x_m$ [here x_m is number of member for x shift] (7)

Shifting y steps (Clockwise): Here we representing each card position by (i,j) and all the points which belongs to y shift are added to an array, size of the array is number of member for the y shifts and k is the array position. We have to determine new position k_n for each card after y shifts.

Suppose randomly selected third point from elliptic curve is (2,1)

Now, P_3(2 mod 30, 1 mod 26) or P_3(2,1)

Now, y = 1

Shifting 1 step

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
i	1	1	2	3	4	5	6	7	8	9	10	11	11	10	9	8	7	6	5	4	3	2
j	1	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1

k_n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
i	2	1	1	2	3	4	5	6	7	8	9	10	11	11	10	9	8	7	6	5	4	3
j	1	1	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1

Now, move card k=0 to new position k_n=1

move card k=1 to new position (0, 0) k_n=2

… … …

… … …

… … …

move card k=21 to new position k_n=0

New position $k_n=(k+y) \bmod y_m$ [here y_m is the number of members for y shift] (8)

After shifting x=2 steps, we get in Matrix (5);

After shifting y=1 steps, we get in Matrix (6);

3.3. Decryption Process

Receiver has public keys $E_p(a,b)$, e_1, e_2 and private key d, and receives from sender final cipher text CCD9C75SC2CJD5D69CD8SJ63, C_0, C_1, C_2 and C_3. Receiver retrieves P_1, P_2 and P_3 from C_0, C_1, C_2 and C_3 using following formula.

$P_1=C_1-d\times C_0$; $P_2=C_2-d\times C_0$; $P_3=C_3-d\times C_0$

Matrix (5). Clockwise shifting X

6C	5C	4C	4D
7C	5D	5S	4S
8C	6D	6S	4H
9C	7D	7S	5H
#C	8D	8S	6H
JC	9D	9S	7H
QC	#D	#S	8H
KC	JD	JS	9H
AC	QD	QS	3H
2C	KD	KS	JH
3C	AD	AS	QH
3D	2D	2S	KH
3S	3H	2H	AH

Matrix (6). Clockwise shifting Y

6C	5C	4C	4D
7C	6D	5D	4S
8C	7D	5S	4H
9C	8D	6S	5H
#C	9D	7S	6H
JC	#D	8S	7H
QC	JD	9S	8H
KC	QD	#S	9H
AC	KD	JS	3H
2C	AD	QS	JH
3C	2D	KS	QH
3D	2S	AS	KH
3S	3H	2H	AH

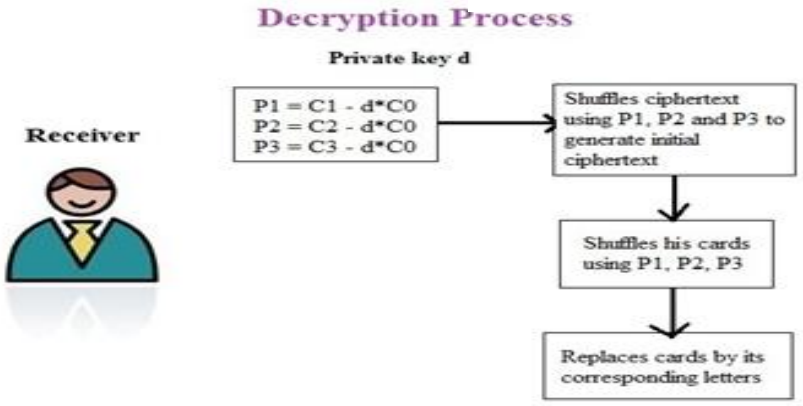


Figure (1c). Decryption process of proposed method

Now receiver arranges final cipher text by 4×(cipher_text_length/4) or 4×(24/4) or 4×6
Position of Matrix (7) is shown on Matrix (8).

Matrix (7). Plain text by using cipher text character

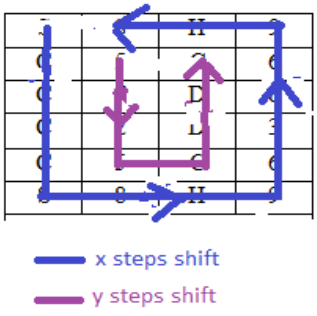
C	C	D	9
C	7	5	S
C	2	C	J
D	5	D	6
9	C	D	8
S	J	6	3

Matrix (8). Position of cipher text

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)
(3,0)	(3,1)	(3,2)	(3,3)
(4,0)	(4,1)	(4,2)	(4,3)
(5,0)	(5,1)	(5,2)	(5,3)

After then from $P_3(x,y)$ receiver shifts same way maintained before x steps and y steps but in anti-clockwise (shown below)

Table (6). Shifting Anti-clockwise



x steps shift contains with those cards which are in the boundary of the matrix. And y steps shift with 2nd and 3rd column except first and last row of the matrix.

Here matrix size 4×6. Number of members of x shift is 2(6+2) or 16 [from eq^n 5] and number of members in y shift is 2(6 - 2) or 2×4 or 8 [from eq^n 6]

[Note that: Number of row is cipher_text_length/4 or 24/4 or 6]

Now our modified elliptic curve point is $P_3(x \bmod x_m, y \bmod y_m)$

Shifting x steps (Anticlockwise): Here we representing each card position by (i,j) and all the points which belongs to x shift are added to an array, size of the array is number of member for the x shifts and k is the array position. We have to determine new position k_n for each card after x shifts.

Suppose we select our third point from elliptic curve is (2,1)

Now, $P_3(2 \bmod 16, 1 \bmod 8)$ or $P_3(2,1)$

Now, x = 2

Shifting 2 steps anticlockwise

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i	0	1	2	3	4	5	5	5	5	4	3	2	1	0	0	0
j	0	0	0	0	0	0	1	2	3	3	3	3	3	3	2	1

k_n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i	0	0	0	1	2	3	4	5	5	5	5	4	3	2	1	0
j	2	1	0	0	0	0	0	0	1	2	3	3	3	3	3	3

Now, move card k=0 to new position k_n=14

move card k=1 to new position k_n=15

… … …

… … …

… … …

move card k=15 to new position k_n=13

New position $k_n=(k+x) \bmod x_m$ [here x_m is the number of members in x shift] (9)

Shifting y steps (Anticlockwise): Here we representing each card position by (i,j) and all the points which belongs to y shift are added to an array, size of the array is number of member for the y shifts and k is the array position. We have to determine new position k_n for each card after y shifts.

Suppose randomly selected third point from elliptic curve is (2,1)

Now, $P_3(2 \bmod 16, 1 \bmod 12)$ or $P_3(2,1)$

Now, y = 1, Shifting 1 step anticlockwise

K	0	1	2	3	4	5	6	7
I	1	2	3	4	4	3	2	1
J	1	1	1	1	2	2	2	2

k_n	0	1	2	3	4	5	6	7
I	1	1	2	3	4	4	3	2
J	2	1	1	1	1	2	2	2

Now, move card k=7 to new position k_n=6

move card k=6 to new position (0, 0) k_n=6

… … …

… … …

… … …

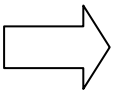
move card k=0 to new position k_n=7

New position $k_n=(k-y) \bmod y_m$ [here y_m is the number of members in y shift] (10)

After shifting x=2 steps anti-clockwise in Matrix (9);
After shifting y=1 steps anticlockwise in Matrix (10):

Matrix (9). X shifting anti-clockwise

D	9	S	J
C	7	5	6
C	2	C	8
C	5	D	3
C	C	D	C
D	9	S	J



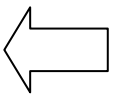
Matrix (10). Y shifting anti-clockwise

D	9	S	J
C	5	C	8
C	7	D	8
C	2	D	3
C	5	C	6
D	9	S	J

Now for $P_2(x,y)$ shifting x steps right and y steps downwards.[reverse of encryption process]
After shifting $P_2(3,5)$ in Matrix (11);

Matrix (11). After shifting of $P_2(3,5)$

5	C	6	C
7	D	8	C
2	D	3	C
5	C	6	C
9	S	J	D
9	S	J	D



Matrix (12). After shifting of $P_1(2,3)$

6	C	5	C
J	D	9	S
J	D	9	S
6	C	5	C
8	C	7	D
3	C	2	D

Now for $P_1(x,y)$ shifting x steps left and y steps upwards.[reverse of encryption process]
After shifting $P_1(2,3)$ in Matrix (12).
Now arrange this matrix row by row for new cipher text: 6C5CJD9SJD9S6C5C8C7D3C2D

Matrix (13). Receiver cards

2H	2C	2D	2S
3H	3C	3D	3S
4H	4C	4D	4S
5H	5C	5D	5S
6H	6C	6D	6S
7H	7C	7D	7S
8H	8C	8D	8S
9H	9C	9D	9S
#H	#C	#d	#S
JH	JC	JD	JS
QH	QC	QD	QS
KH	KC	KD	KS
AH	AC	AD	AS

Matrix (14). After process get cards

6C	5C	4C	4D
7C	6D	5D	4S
8C	7D	5S	4H
9C	8D	6S	5H
#C	9D	7S	6H
JC	#D	8S	7H
QC	JD	9S	8H
KC	QD	#S	9H
AC	KD	JS	3H
2C	AD	QS	JH
3C	2D	KS	QH
3D	2S	AS	KH
3S	3H	2H	AH

Now to find out which text inside of each two cards lets shuffle receiver card using P_1, P_2 and P_3 on Matrix (13). The process is same as encryption process. After doing all the process we get shuffled cards shown by on Matrix (14).
Now place each character between two cards, the table is given below on Matrix (15).
Now replace each pairs of cards (from new cipher text) by letter between them.
New cipher text = 6C5CJD9SJD9S6C5C8C7D3C2D on Table (7).

Matrix (15). Character between two cards

6C	A	5C	N	4C	0	4D
7C	B	6D	O	5D	1	4S
8C	C	7D	P	5S	2	4H
9C	D	8D	Q	6S	3	5H
#C	E	9D	R	7S	4	6H
JC	F	#D	S	8S	5	7H
QC	G	JD	T	9S	6	8H
KC	H	QD	U	#S	7	9H
AC	I	KD	V	JS	8	3H
2C	J	AD	W	QS	9	JH
3C	K	2D	X	KS		QH
3D	L	2S	Y	AS	,	KH
3S	M	3H	Z	2H	.	AH

Table (7). Pairs of Cards

Card 1	Card 2	Letter
6C	5C	A
JD	9S	T
JD	9S	T
6C	5C	A
8C	7D	C
3C	2D	K

Now receiver gets plain text “ATTACK”
End of decryption process shown on Table (8).

Table (8). End of decryption process

2H	A	2C	N	2D	0	2S
3H	B	3C	O	3D	1	3S
4H	C	4C	P	4D	2	4S
5H	D	5C	Q	5D	3	5S
6H	E	6C	R	6D	4	6S
7H	F	7C	S	7D	5	7S
8H	G	8C	T	8D	6	8S
9H	H	9C	U	9D	7	9S
#H	I	#C	V	#d	8	#S
JH	J	JC	W	JD	9	JS
QH	K	QC	X	QD		QS
KH	L	KC	Y	KD	,	KS
AH	M	AC	Z	AD	.	AS
XH	x1	XC	x2	XD	x3	XS

Introducing New Characters

Here new cards XH, XC, XD and XS can be introduced for new characters x1, x2 and x3. The existing formula is same as before, no need to change the formula to introduce new characters. If we don’t need any character we can leave it as “null” or white space.

Finally, as our proposed algorithm selects three random points for every message it provides differently shuffled cards in every single message. Thus we can say our mapping is dynamic.

4. Summary

Encryption:

Step 1: Sender receives public key $E_P(a,b)$, e_1 and e_2 from receiver.

Step 2: Sender creates an elliptic curve using $E_P(a,b)$ and randomly selects three points (P_1, P_2 and P_3) over elliptic curve.

Step 3: Sender shuffles his playing cards (4×13 matrix) using P_1, P_2 and P_3.

Step 4: Sender puts each character between two cards which is shuffled in step 3.

Step 5: Sender replaces each character by two cards from its left and right, and creates initial ciphertext.

Step 6: Sender shuffles his initial ciphertext using P_1, P_2 and P_3 and creates final ciphertext

Step 7: Sender encrypts randomly selected point P_1, P_2 and P_3 using idea of ElGamal Cryptosystem and gets C_0, C_1, C_2 and C_3.

Step 8: Sender sends C_0, C_1, C_2 and C_3 and final ciphertext to the receiver

Decryption:

Step 1: Receiver decrypts C_0, C_1, C_2 and C_3 and gets P_1, P_2 and P_3 using his private key d.

Step 2: Receiver reversely shuffles final ciphertext using P_1, P_2 and P_3 and gets initial ciphertext.

Step 3: Receiver shuffles his playing cards using P_1, P_2 and P_3,

Step 4: Receiver puts each character between two cards which is shuffled in step 3.

Step 5: Receiver replaces each two cards of initial ciphertext by character between them.

Step 6: Completing replacement of all the cards from initial ciphertext receiver gets plaintext.

Simulation Result Simulation is done by using Toshiba Satellite C600 with system configuration dual core processor @2.30 GHz and 2 GB Ram. Word length= 409, Text Length: 2325

Table (2a). Simulation result for encryption and decryption time

Key size	Word Counter	Encryption Time (s)	Decryption Time (s)
128 bit	409	0:825	0:463
163 bit		0:825	0:473
192 bit		0:857	0:452
256 bit		0:846	0:440
384 bit		0:846	0:441
512 bit		0:870	0:464

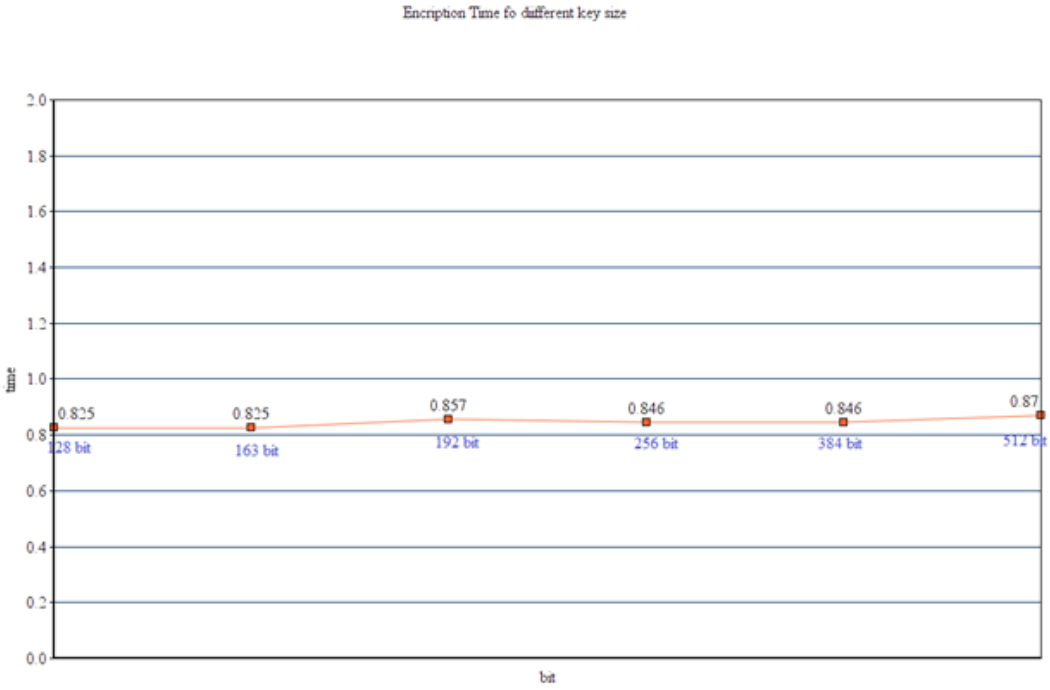


Figure (2a). Simulation Graph for Time (Encryption)

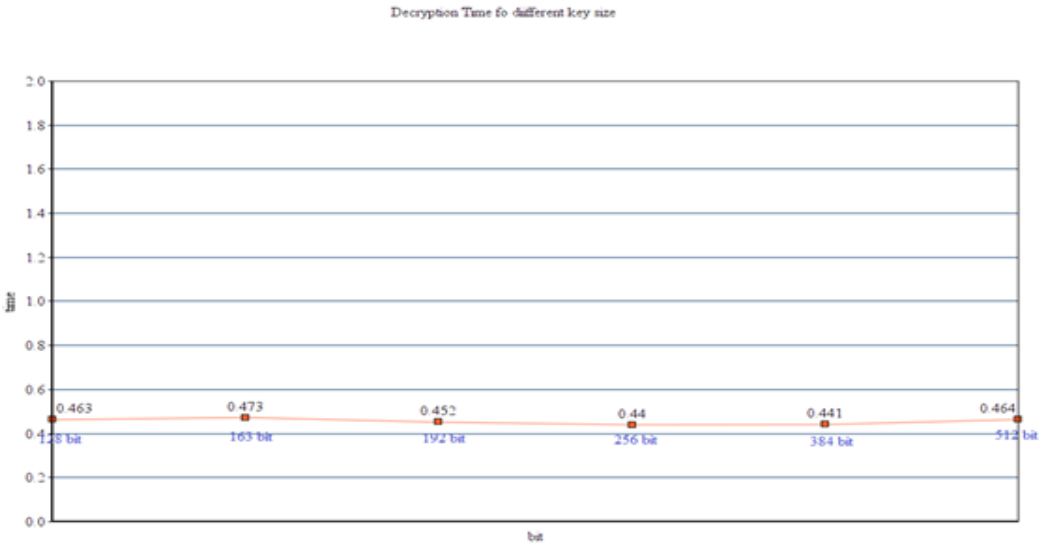


Figure (2b). Simulation Graph for Time (Decryption)

From table (2a), figure (2a) and (2b) we can see that bit size does not affect our algorithms encryption and decryption time.

Comparison As our proposed of algorithm requires mapping, we compared our algorithm with those algorithm which requires mapping. Implementation of Elliptic Curve Cryptography on Text and Image [7] and Implementation of Text based Cryptosystem using Elliptic Curve Cryptography [6] simulation was performed using Lenovo ideapad Z510 laptop with system configuration of i7 processor @2.20GHz and 8GB Ram [6]. Our simulation is done by using Toshiba Satellite C600 with system configuration dual core processor @2.30 GHz and 2 GB Ram. Word length=409, Text Length: 2325. We have done our comparison with only those methods where mapping is required.

Table (3a). Simulation result of referenced method

Method	Word length	Encryption Time	Decryption Time	Mapping
Implementation of Elliptic Curve Cryptography on Text and Image	1	0.2 seconds	0.3 seconds	Required
Implementation of Text based Cryptosystem using Elliptic Curve Cryptography	409	1.95 seconds	0.83 seconds	Required
Our Algorithm	409	0.857 seconds	0.452 seconds	Required

Though our simulation environment is better than referenced environment, our algorithm performs better then referenced methods.

Simulation Result of Image Encryption/Decryption

The implementation is performed on i7 CPU 2.20GHz lenovo laptop with 8 GB RAM using Mathematica version 10. The Elliptic curve used here is the 512 bit Standard Elliptic curve given by Elliptic Curve Cryptography.

$$y2 = \{x3 + ax + b\} \bmod [p] \tag{11}$$

where

p = 8948962207650232551656602815159153422162609644098354511344597187200057010413552439917934304191956942765446530386427345937963894309923928536070534607816947;

a = 6294860557973063227666421306476379324074715770622746227136910445450301914281276098028027990968407983962691151853678563877834221834027439718238065725844264138;

b = 3245789008328967059274849584342077916531909009637501918328323668736179176583263496463525128488282611559800773506973771797764811498834995234341530862286627;

G = {6792059140424575174435640431269195087843153390102521881468023012732047482579853077545647446272866794936371522410774532686582484617946013928874296844351522, 6592244555240112873324748381429610341312712940326266331327445066687010545415256461097707483288650216992613090185042957716318301180159234788504307628509330};

n = 8948962207650232551656602815159153422162609644098354511344597187200057010413418528378981730643524959857451398370029280583094215613882043973354392115544169;

Here n is the cyclic order of the Elliptic curve with G as generator

nB = 96192759682482119853328425949563698712343813919172976158104477319333745612481875498805879175589072651261284189679678167647067832308974867524089740051333;
Here *nB* is the private key of the receiver.

Pb = {155909306574095688254303186081739466582364593248005646967432362224511343712118043139025951742310192095684266368225423091074452980008684932415984684310104 9,268762854425619391532292656002566976899442050795167352328551987675795436125123497395455936256239827381890777120258304469374304988963657760697200655197567 1};
Pb is the public key of the receiver, which is formed using the private key of the receiver, given by point multiplication of *nB* and *G*.

nA = 94268904488832477456261857430572424738096937640789516634942387772947070700232237988829761592077291198236058505886084604294126475673608974091172098560224 01;
nA is the private key of the sender

Pa = {77511187111048294650456399420708073707837290480871566989671986079254879520158149804269980291496112687534710426194837742349300715457321680491523551899648 49,58735024067276542220759198140648261016906441523144403900269298371649521237910226239943370030592982018358818460509976413039545990712468144658861929061944 93};
Pa is the public key of the sender.

The sender sends the cipher image along with the Digital Signature calculated with the pixels value of the cipher image. The public key of the receiver was used to encrypt the image and the private key of the sender was used to provide digital signature to the cipher image being sent.While decrypting the cipher image, the receiver uses his private key to decrypt the cipher image. Authenticity provides the proof that the message came for the intended sender and integrity validates that the message was not altered or changed during the transit. An altered cipher image will generate.

5. Conclusions

Data confidentiality is important to protect data from unauthorized access of confidential information. In this research work we have implemented a new way for text encryption and decryption using card shuffling logic which provides data confidentiality. For higher security with lesser key size, we use Elliptic curve in our research work.

In our research work we have created initial cipher text by shuffling and mapping with playing cards. These cards are shuffled using three randomly selected points over elliptic curve. We have implemented double encryption process by shuffling the initial ciphertext using those randomly selected points and generate final ciphertext for more security.

From the performance table, comparison table and security analysis section we can observe that our proposed algorithm has a lot of positive aspect. As Elliptic Curve Cryptography provides equal security like other cryptographic system but with less key size, it is very suitable for devices which have power, storage and processing limitation. So, our algorithm can be implemented in that device where less processing power and energy requires.

REFERENCES

[1] William Stallings, ―Cryptography and Network Security, Principles and Practice‖, [Forth Edition], Pearson Education Inc., 2006.

[2] Behrouza A. Forouzan and Debdeep Mukhopadhyay, ―Cryptography and Network Security‖, [Second Edition], McGraw Hill Education Private Limited, 2008.

[3] Georgios Loukas and Gulay Oke, ―Protection against Denial of Service Attacks: A Survey‖, Oxford University Press, 2009.

[4] Laiphrakpam Dolendro Singh and Khumanthem Manglem Singh, Implementation of Text Encryption using Elliptic Curve Cryptography, ELSEVIER, ScienceDirect, Procedia Computer Science 54(2015) 73-82 (2015).

[5] D. Sravana Kumar, CH. Suneetha and A. ChandrasekhAR, Encryption of Data using Elliptic Curve over Finite Fields, IJDPS Vol.3, No.1, January 2012.

[6] S. Maria Celestin Vigila and K. Munesswaran, Implementation of Text based Cryptosystem using Elliptic Curve Cryptography, International Conference on Advance Computing, IEEE, pp. 82-85, December (2009).

[7] Megha Kolhekar Anita Jadhav Implementation of Elliptic Curve Cryptography on Text and Image, International Journal of Enterprise Computing and Business System, vol. 1, issue 2, July (2011).