Elliptic Curve Cryptography for Secured Text Encryption

Keerthi K
Department of CSE
National Institute of Technology
Puducherry India
keerthikamalakshan@gmail.com

Dr.B.Surendiran
Department of CSE
National Institute of Technology
Puducherry India
surendiran@gmail.com

***Abstract*—Elliptic Curve cryptography is a public key cryptographic system where the message is encrypted using private key of sender and decryption is done using senders public key and the receiver's private key. This paper introduces a new mapping technique for encoding the message into affine points on the elliptic curve. Mapping technique convert the plain text into ASCII values and then convert this into HEXADECIMAL. The converted Hex values are grouped together to form the x and y coordinates. The converted values are encrypted in reverse order to prevent security attacks. This method reduce the overhead of common look up table shared between the sender and the receiver. Also it avoids the extra padding bits when group count is odd, which can be considered as NULL values.**

Keywords—Elliptic curve cryptography (ECC); Public key cryptography; Mapping; Encryption/Decryption.

I. Introduction

Cryptography is the method of analysis and implementation of some mathematical concepts that are used for secure communication channels in the presence of intruders. The major part of cryptography is encryption and decryption. Encrypted message contain all the information of the original transaction but not in human readable format. Common cryptographic primitives are RSA, ECC, hash function and digital signature etc.

Elliptic curve cryptography (ECC) was discovered in 1985 by Neal Koblitz and Victor Miller .The main advantage of elliptic curve cryptography is that it provides better security with smaller key size. For example, 160 bit elliptic curve provide same security as that of a 1024 bit RSA .It is also difficult to solve discrete logarithmic problems on elliptic curve. The elliptic curve over prime field comprises of coefficients of the elliptic curve and the base point, which is a point on the curve. The curve selected showed be known to both sender and the receiver .The major operation in ECC is scalar multiplication which comprises of Point addition and Point doubling.

The encryption is by mapping the message into points on the curve and then performing scalar multiplication using sender's private key. After decryption the points are again decoded into the actual message.

In this paper we have introduced a secured mapping technique that maps the message to affine points on the curve. Our technique avoid the overhead of sharing the common look up tables between the sender and the receiver. The reverse order encryption provides more security for the text encryption.

The plaintext are mapped into affine points on the elliptic curve by using the proposed method. The encryption is performed on these points. Let the curve parameter be 192 bit, then the ASCII values are mapped into 192 bit x and y. These points are encrypted using scalar multiplication algorithm. Here we have opted binary NAF point scalar multiplication method. The steps for encryption and decryption are included in the following section.

II. PRELIMINARIES

In this section we summarized the basic curve parameter and operations on a finite field curve E(F_p) where p is a prime number .The basic operation on a finite field curve are point addition and point doubling on affine points. Here we have introduced Projective coordinates which is obtained from affine points. Projective coordinate points helps to reduce the number of inversions, is explained in the following section.

The elliptic curves E over a field is defined as:

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

Equation (1) is called *Weierstrass equation.* An elliptic curve E over a finite field GF(F_p) is the simplified form of the Weierstrass equations which is:

$$y^2 = (x^3 + ax + b) mod\ p \tag{2}$$

Where a, b ε K .The discriminant of this curve is:

$$\Delta = (4a^3 + 27\, b^2) \bmod p \neq 0 \quad (3)$$

The set of points on the elliptic curve along with a special point ∞, called the point at infinity, form a group under addition. The identity element of the group is the point at infinity.

The basic operations on the elliptic curve are Point Addition and Point Doubling which forms the building blocks of scalar multiplication. All the basic operations are performed in Projective coordinates in order to reduce the costly inversion operation.

A. Point Addition

Let P and Q be the two points on the curve ,the addition of P and Q results in P+Q. i.e. R=P+Q .The basic operation of addition of points is by drawing a line through the points P and Q. This line intersect a point on the curve, i.e. -R. The inverse of this point is the actual result R.

B. Point Doubling

Let P be a Point on the curve ,Doubling results in R=2P.The computation is pictorially given by drawing a tangent on the point P. The Point obtained by intersection of this line is -R. The inverse of this point result in the actual point R.

Algorithm 1: Point Doubling ($y^2 = x^3 + ax + b$), Jacobian Coordinates

```
1. If $P = \infty$ then
2.     Return ($\infty$)
3. $B \leftarrow 4 * X_1 * Y_1^2$
4. $A \leftarrow 3 * X_1^2 + a * Z_1^4$
5. $X_3 \leftarrow A^2 - 2 * B$
6. $Y_3 \leftarrow A * (B - X_3) - 8 * Y_1^4$
7. $Z_3 \leftarrow 2 * Y_1 * Z_1$
8. *Return* $(X_3 : Y_3 : Z_3)$
```

C. Point Inversion

The inverse of a give point P =(x1, y1) on a finite field curve is given by the point (x1,-y1) which can be called as -P.

D. Projective Coordinates

In [1] the authors have described the use of Projective coordinate system. Here two dimensional affine points are converted into three dimensional Projective Coordinates. This has considerably reduced the number of inversions.

If *(x, y)* be the affine point on the curve, then the Projective Coordinate of this point can be denoted as *(X,Y,Z)* and $Z \neq 0$.The projective form of the Weierstrass equations can be obtained by replacing x with X/Z^c and y by Y / Z^d and clearing the denominators.

Algorithm2: Point Addition on Affine-Jacobian Coordinates

```
1. If $Q = \infty$ then *Return* $(X_1 : Y_1 : Z_1)$
2. If $P = \infty$ then *Return* $(x_2 : y_2 : 1)$
3. $T_1 \leftarrow Z_1^2$
4. $T_2 \leftarrow T_1 * Z_1$
5. $T_1 \leftarrow T_1 * x_2$
6. $T_2 \leftarrow T_2 * y_2$
7. $T_1 \leftarrow T_1 - X_1$
8. $T_2 \leftarrow T_2 - Y_1$
9. If $T_1 = 0$ then
10.     If $T_2 = 0$ then
11.         Use algorithm 1 to compute $(X_3 : Y_3 : Z_3) = 2(x_2 : y_2 : 1)$
            *Return* $(X_3 : Y_3 : Z_3)$
12. Else *Return* ($\infty$)
13. $Z_3 \leftarrow Z_1 * T_1$
14. $T_3 \leftarrow T_1^2$
15. $T_4 \leftarrow T_3 * T_1$
16. $T_3 \leftarrow T_3 * X_1$
17. $X_3 \leftarrow T_2^2$
18. $X_3 \leftarrow X_3 - T_4$
19. $X_3 \leftarrow X_3 - 2 * T_3$
20. $T_3 \leftarrow T_3 - X_3^2$
21. $Y_3 \leftarrow T_2 * T_3$
22. $T_4 \leftarrow Y_1 * Y_4$
23. $Y_3 \leftarrow Y_3 - T_4$
24. *Return* $(X_3 : Y_3 : Z_3)$
```

The most commonly used projective coordinate system are:
Standard Coordinate: c= 1 and d = 1
Jacobian Coordinate: c= 2 and d = 3
Lopez-Dahab (LD) Coordinate: c=1 and d=2

In this paper all the implementations are done in Jacobian coordinates, where the point to infinity ∞ corresponds to (1 : 1 : 0) and the negative of (X : Y : Z) is given as (X : -Y : Z).

Algorithm 1 defines the Point Doubling on Jacobian coordinates, Results in 2P where P ε (F_p) and also Algorithm 2 defines Point Addition results in P + Q ,where P ε (F_p) and Q ε (F_p) and P in Jacobian coordinate and Q in affine coordinate. Book [1] gives the detailed explanation of the algorithms.

E. Encryption/Decryption

Elliptic Curve cryptography is a public Key cryptosystem we need both private key and public key. Let Alice and Bob are the communicating parties, then they should have a secured channel for communication between them. So this they need secret keys for converting the actual message into known readable format. Key generation is the important part of ECC, so we need both public key and private key. Alice will encrypt the message using Bob's public key and Bob will decrypt the message using his private key.

For Elliptic Curve Cryptography both the sender and the receiver should know the chosen Elliptic curve $E(F_p)$.The key Generation step is as follows:

Select a constant **d** ε (0 to n-1), where n is the maximum limit, which is a prime number. Generate public key Q:

$$Q = d * P \quad (4)$$

Where P is the Point on the curve and P is the private key.
Encryption:
Message **m** is transformed into affine point **M** on the curve.
Select **k** ε (0 to n-1).The plain text is transformed into two cipher text C1 and C2 and send these both cipher text.

$$C1 = k * P \quad (5)$$

$$C2 = M + k * Q \quad (6)$$

Decryption:
The **d** used for generating Public key is used for decryption. Decryption will returns the actual message M from the cipher text C1 and C2.

$$M = C2 - d * C1 \quad (7)$$

III. RELATED WORK

In the literature survey we have highlighted some of the methods that has been used for secure encryption in elliptic curve cryptography. In 1985,Neal Koblitz [2] introduced public key cryptosystem ,elliptic curve over finite field which uses the multiplicative group that is more secured. Also states that the discrete logarithmic problem on elliptic curve make it more harder hence it is more secured compared to other public key cryptosystems. Victor Miller in [3] states Diffie-Hellman Key exchange protocol that is secured from malicious attacks. The elliptic curve groups also provides same security with smaller key size, which can be used in small embedded devices with less space and helps in lower power consumption. Hackerson introduced the basic building block for the implementation of elliptic curve cryptography in the book [1].Also he included varies scalar multiplication and affine projective coordinate systems and also includes the NIST primes reduction method for varies NIST primes.

Mapping method introduced in [4] shows how to map a plaintext into affine points on the curve ,by converting the message into its ASCII values and these integers are converted into affine points by multiplying the ASCII value with a base point on the curve.[5] explains decoding using koblitz method, where the message is converted into ASCII values and these are transformed into x and y coordinate on the curve by using Koblitz's method. By taking x=mk+1 where k is a constant and repeat a step until we get y coordinate on the curve.

The method proposed by Amounas in [6] states the mapping of plaintext into affine points on the curve by using matrix method .Where the alphanumeric characters is mapped into points on the curve and these points are mapped as the matrix coordinates by multiplying it with a non-singular matrix A having only integers. While decoding the inverse of matrix A is taken for the mapping of the cipher text into the actual message. A is chosen such that determinant of A = ± 1.In 2015 [7]. Sigh introduced a method of mapping by grouping the ASCII values to form the big number and these points used for encryption. [8] Paul used the same method as in [5] for mapping the plaintext to affine points on the curve, but uses columnar transposition on the mapped values in-order to provide security. In [9] author have introduced a new method for fixed and variable size mapping, by introducing an initial vector. Based on the initial vector the same text message is mapped onto various points on the elliptic curve.

IV. METHOD

In this paper we proposed a new method for mapping the plaintext into affine points on the curve. In this method the main idea is that convert each character of the plaintext into ASCII values and the ASCII values are again converted into HEXADECIMAL values and then perform the grouping of these HEXADECIMAL values based on the size of the curve parameter. For example if we are using 192 bit NIST prime then the HEXADECIMAL is grouped to 192 bit (x, y) coordinate. But the point is that these groups perform the scalar multiplication in the reverse order of the HEXADECIMAL input obtained or either you can perform any columnar transposition for the input points x and y in order to provide security. After mapping encryption is performed. In this implementation we have used binary NAF method [1].

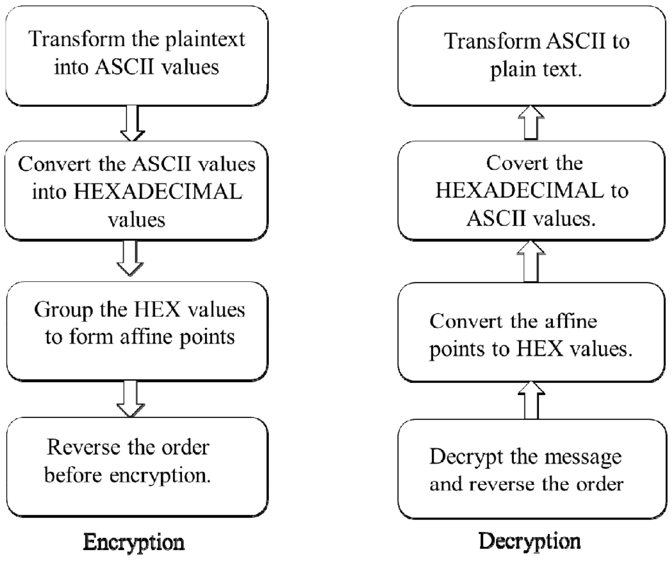
A. *Proposed Method*

The proposed method shows a new mapping technique for mapping the message into affine points on the curve. For performing Elliptic curve encryption and decryption both the sender and the receiver must the following:

- The Elliptic curve chosen $E(F_p)$.i.e. For finite field curve : $y^2 = x^3+ax+b \pmod p$,the parameters are a, b and p where p is prime number.
- The base point with order N.
- The order in which the mapped HEXADECIMAL encryption is performed.

The mapping method proposed in this paper convert Message M into ASCII values and converting these ASCII into HEXADECIMAL and grouping of HEXADECIMAL into finite set of input size. Reverse the order of the HEXADECIMAL value obtained. These results are encrypted using Scalar multiplication algorithm and convert it into cipher text.

The figure [1] shows the block diagram of the proposed method used for encryption of the text using ECC. The first step is key generation generate the public key and the private key. The public key and the private key generated is used for encryption. The decryption is done using the **d** value chosen.


Transform the plaintext into ASCII values
Convert the ASCII values into HEXADECIMAL values
Group the HEX values to form affine points
Reverse the order before encryption.
Encryption
Transform ASCII to plain text.
Covert the HEXADECIMAL to ASCII values.
Convert the affine points to HEX values.
Decrypt the message and reverse the order
Decryption

V. IMPLEMENTATION

This section includes the implementation of proposed method. All the input for the ECC encryption is chosen as 192 bit NIST prime parameters. Since we are using HEXADECIMAL values of the plaintext, we have represented the values of a, b, p and the base point (x1,y1) is HEXADECIMAL format. For example the values chosen in our method are as follows:

$$P_{192} = 2^{192} - 2^{64} - 1 \quad (8)$$

a:
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFF FFC
b:64210519E59C80E70FA7E9AB72243049FEB8DEECC14 6B9B1
x:D458E7D127AE671B0C330266D246769353A012073E97 ACF8
y:325930500D851F336BDDC050CF7FB11B5673A1645086 DF3B
d:A78A236D60BAEC0C5DD41B33A542463A8255391AF6 4C74EE
k:C4BE3D53EC3089E71E4DE8CEAB7CCE889BC393CD8 5B972BC

A. *Plain Text*

Let the message to be transmitted is:
#)652323111656565KSHGDSJHDSJKJKJKaFHJHJ v^8888****/14236

B. *Mapping*

Convert each input symbols into ASCII values. Then ASCII values are converted into HEXADECIMAL values.

$$\# \rightarrow 35 \rightarrow 23 \quad (9)$$

Using method in equation above we can convert the whole plaintext into its equivalent HEXADECIMAL format. Thus the plaintext can be transformed as:
Hexadecimal:
2329363532333233313131363536353635254B53484744534 A4844534A4B4A4B4A4B6146484A484A765E383838382A 2A2A2A2F313432333600
These Hexadecimal values are grouped to form the x and y coordinate as 192 bit input x and y.
x1:
2329363532333233313131363536353635254B534847445
y1:
34A4844534A4B4A4B4A4B6146484A484A765E383838382 A
x2: 2A2A2A2F313432333600
y2: NULL

The encryption is done by considering (x2, y2) as one and (x1,y1) as second.

C. *Encryption*

The encryption is done with the affine points obtained in the mapping step.

Perform Q=d*P and C1=k*P, where P is affine point (x, y) The cipher text C2 can be formed by adding the Message M=(x1,y1) with k*Q using the algorithm given in [2].

The cipher text obtained from the plaintext is:
cx1=99FD28A5A8D71B5942438F004BCA838CEC155DAA DFBDE64A
cy1=6DF4A57AA0DCFD1C31F092AF52107C150E1EB920 B7082B21
cx2=CE19BF6AA317DBE34DB07377B4A557E3844CB0D6 8018B879
cy2=ADF6DC2F009BC60125C457C854932AE419CB69D30 AB4EF8C
This cipher text is transmitted as

$$\text{Transmit}[C1, M + k * Q] \quad (10)$$

D. *Decryption*

The message can be decrypted by the equation (7).Then the obtained affine points are mapped back to the message using proposed mapping method as shown in figure [1].

The Decrypted affine points are:
x5:
2329363532333233313131363536353635254B534847445
y5: 34A4844534A4B4A4B4A4B6146484A484A765E383838 382A

x5: 2A2A2A2F313432333600
y5: NULL

The Message after performing proposed mapping method is: #)652323111656565KSHGDSJHDSJKJKJKaFHJHJv^8888****/14236

From the result it is clear that we need not have a common look up table for both the sender and the receiver. Sender and receiver should know the Elliptic curve chosen and also the order in which encryption is performed. It is easy to determine the message if ASCII character is obtained. So in-order to prevent the attack we have converted the ASCII to hexadecimal .Hence attacker will have a misunderstanding about the message. Also the message is encrypted in reverse order values which help provide security. The main advantage of this method is that we don't have to pad extra bits if the grouped hexadecimal number is odd in number. Since we are using Hexadecimal, the input with length zero is taken as **NULL**.

VI. Analysis

All the implementation is done on Lenovo G580, with Intel i3, 2GB RAM and using 192 bit NIST prime recommended elliptic curve parameter [10]. The implementation is done using C.

The input can be of any size. Table 1 shows the time taken for the encryption and the decryption for the given sized message.

The result shows that, encryption takes more time than decryption, since it contain more number of scalar multiplications. The method implemented does not require any mapping prior to encryption and also no common look-up table is needed.

TABLE 1: Time Taken for encryption/Decryption

Time Taken(in secs)	**Operations**	
	Encryption	***Decryption***
Our method	0.08	0.06
Reference[7]	0.09	0.10
Reference[4]	1.95	0.83

VII. CONCLUSION

Elliptic Curve Cryptography is a public key cryptographic system .ECC provides more security than RSA with lesser key size. The security of ECC is more since it has discrete logarithmic problem, which is difficult to solve.

Before performing text encryption we need to map the text into points on the curve. In this paper, we have implemented a new mapping technique that maps the plaintext into affine points on the chosen elliptic curve. In the proposed method, the characters in the plaintext are converted into ASCII values and the next step is to convert it into HEXADECIMAL. The entire HEXADECIMAL values are grouped based on the input size. The encryption is done in the reverse order of the HEXADECIMAL result in-order to provide security.

ECC implementation is more effective compared to RSA for key size and also security. It provide more security with lesser key size. So this can be utilized in electronic devices with lesser memory and lower power consumption. The security in the mapping technique will provide double security for the text encryption. Here we have implemented a secure mapping technique along with less overhead in mapping.ie, our implementation will help to reduce the common look-up table between the sender and the receiver and hence reduce the overhead. So the encryption procedure works faster.

One of the major advantage of this method is that we don't have to pad extra bit when the grouped hexadecimal is odd in number. Because the length zero group is taken as NULL.

References

[1] D,Hakerson, S.Vanstone, and A.J Menezes, “Guide to Elliptic Curve Cryptography ” , 2004

[2] N. Koblitz, “Elliptic Curve Cryptosystems”, Mathematics of Computation, vol. 48. no. 177,pp. 203-203,1987.

[3] V.Miller ,”Use of Elliptic Curve in Cryptograhy”, Advances in Cryptography CRYPTO’85, vol.LNCS 218,PP.417-126,1986. K. Elissa, “Title of paper if known,” unpublished.

[4] S .M . C Vigila and Munseeswaran,”Implementation of Text based cryptosystem using elliptic curve cryptography”, 2009 1st International conference on advances computing ,ICAC 2009,pp.82-85,2009.

[5] P.Bh,D.Chandeavathi , and P .P. Roja,”Encoding and Decoding of a message in the Implementation of Elliptic Curve Cryptography using Koblitz’s Method”, International Journal on computer science ,vol.02,no .05,pp-1904-1907,2010.

[6] F.Amounas and E. H. E. Kinani,”Fast Mapping Method based on Matrix Approach For Elliptic Curve Cryptography”, vol. 1, no. 2, pp. 54-59,2012.

[7] L. D. Singh and K. M. Singh,”Implementation of Text Encryption using Elliptic Curve Cryptography”, Procedia Computer Science,vol.54, no.1, pp. 73-82,2015

[8] C. T. M and V. Paul, “Secure Method for embedding plaintext on an elliptic curve using TDMRC code and Koblitz method”, Journal of Theorectical and Applied Information Technology,vol. 84,no. 2,pp. 298-304,2016.

[9] J. Muthukuru, B. Sathyanarayana, “Fixed and Variable Size Text Based Message Mapping Techniques using ECC”, vol. 12,no. 3, 2012.

[10] www.csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTeCur.pdf, 1999