IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-3, March 2019 E-ISSN: 2347-2693

A Method of Text Message Mapping in Elliptic Curve Cryptosystems

T. K. Ghosh

Dept. of Computer Science, Bankura Sammilani College, P.O: Kenduadihi, Dist: Bankura(WB), India

Corresponding: tapas.bsc38@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i3.395398 | Available online at: www.ijcseonline.org

Accepted: 11/Mar/2019, Published: 31/Mar/2019

***Abstract*:** Use of web applications in our day to day life are increasing, which result in an increase in the amount of sensitive data transmitted over the Internet. Sensitive data can be transmitted securely over the Internet by encrypting them. Among several public key cryptosystems elliptic curve cryptosystem is a recent one. This cryptosystem gives us stronger security with smaller key size, thus making it useful in those devices which have limited memory and power consumption ability. Encoding of text messages into elliptic curve points and decoding encoded points into original plaintext is always challenging in ECC. General approach for message mapping is to encode the characters of a message into the x-Coordinate of a point on an elliptic curve E_p (a, b). Then find out corresponding y value so that (x, y) lies on E_p (a, b). The point is then encrypted and transmitted. Since for every point both x and y values are to be transmitted and for stronger security, value of p is of at least 160 bits in today's standard, therefore a major concern is to diminish the number of bits used in mapped point (x, y). In this paper we will discuss about a new mapping methodology of a group of alphanumeric characters into an elliptic curve point which reduces the number of bits to be transmitted without compromising the data security.

Keywords: Elliptic Curve, Mapping algorithm, encoding, decoding, encryption, decryption.

I. INTRODUCTION

Cryptography is a practical means for transmitting sensitive data through insecure channel. Among several public key cryptosystems Elliptic Curve Cryptosystem is a crucial one. In mid 1980's Elliptic curve cryptosystem was independently proposed by Victor Miller [1] and Neil Koblitz [2]. Initially ECC was not so popular. But with time its popularity is increasing. Smaller key size, hardness for solving ECDLP, low power consumption etc make ECC suitable for use in devices like mobile phones which have limited storage and processing capability. Almost all cryptographic systems need a method of converting plaintext characters into a numerical value which can be used to do mathematical computation. In ECC we also have a need of mapping plaintext messages into points on an Elliptic Curve. The points produced thus are then used to generate other points on the same Elliptic Curve by the add-double-multiply rules [3]. These points are then used as cipher text. To achieve strong security the minimum bitlength of each ciphertext point must be at least 320 bits. So a common goal is to reduce the number of ciphertext points. In this paper we propose a message mapping algorithm which takes a substring of 9 characters at a time, and generate a point on an Elliptic Curve which reduces the number of ciphertext points. After this brief discussion the remaining parts of this paper is organized as follows: In Section II some related works along with a brief description of elliptic curve cryptography over a prime field are discussed. The proposed scheme of simple text messages encoding into EC points, and decoding from points to original message is discussed in section III. In section IV a comparison of results obtained from the proposed scheme and other methods are discussed. Section V concludes the paper.

II. RELATED WORK

The simplified Weierstrass form of a non super singular elliptic curve E_p(a, b) over a Galois field GF(p) (p>3 and is prime) [4] is defined by the equation:

$$E: y^2 = x^3+ax+b \pmod p$$

where a, b $\in z_p$ and $\Delta = 4a^3+27b^2 \neq 0 \bmod p$.

In elliptic curves, multiplying a point P with an integer n, can be implemented as, successive n-1 times addition of P with itself. This is called nP operation. Using add double rule this can be done efficiently. For large value of n computing nP is easy but it is hard to compute n from Q = nP. This is the Elliptic Curve discrete logarithm problem. It is also believed that ECDLP is harder than DLP to solve [5]. Elliptic curve point addition and doubling formulae for points P(x_1, y_1) & Q(x_2, y_2) are as follows: $x_3 = (s^2-x_1-x_2) \bmod p$ and $y_3 = (s(x_1-x_3)-y_1) \bmod p$ where

$$s = \begin{cases} \frac{y_2-y_1}{x_2-x_1} \ mod\ p\ ;if\ P \neq Q\ (point\ addition) \\ \frac{3{x_1}^2+a}{2y_1} \ mod\ p\ ;\ if\ P = Q\ (point\ doubling) \end{cases}$$

The points on $E_p(a, b)$ forms a finite abelian group where identity element is the point at infinity. Two parties may exchange data securely by elliptic curve analogue of the Massey-Omura system or elliptic curve analogue of the ElGamal system [2].

In Massey-Omura system a publicly known elliptic curve E over a prime field GF(p) is taken. Let O = order (E). Suppose user Allice wants to send user Bob a message m. She first converts m into a point P on E, then take a random integer α with $0 < \alpha < O$ satisfying gcd (α, O) =1 and transmits αP. Similarly Bob also take a random integer β with same properties. He then compute β(αP) and sent it to Allice. Alice now computes α^{inv} (βα)P and gets βP because α^{inv} α ≡ 1 mod O . She sends it to Bob. Bob upon receiving βP computes β^{inv}(βP) = P. Decoding P Bob gets the original massage.

In ElGamal system of elliptic curve cryptography Allice first obtains P from m. Also a point G is chosen which is publicly known. The receiver Bob take α randomly and make αG public keeping α secret. Allice now send Bob a pair of points (βG, P + β(αG). Receiving the pair of points Bob multiplies 1st point βG by α then subtracts it from 2nd point to get P. He now decodes P and get the original message m.
In both of the above systems we see that encryption and decryption starts and ends with mapping the plaintext message into points on an elliptic curve. Several mapping methods exist to convert plaintext characters into elliptic curve points. Some of these algorithms are discussed as follows:

Method – 1: Algorithm of this method [4] takes a base point G (The point whose order is equal to the order of the elliptic curve) on $E_p(a, b)$ as input, multiply it with the ASCII value of each plaintext character to get a point on $E_p(a, b)$. For example ASCII value of 'c' is 99. So to map 'c' on E_p (a, b) this algorithm computes 99*G. It is a common method generally used where the time taken to encode is considered rather than the security in using it. Note that it is easily vulnerable to a frequency analysis attack so it is a less secure method.

Method – 2: In [6] the authors have used a non singular matrix to map same characters in the message to different points on an elliptic curve. It is a combination of method-I and a matrix to permute the position of the elliptic curve points. According to similar scheme as mentioned in method-I plaintext characters are mapped in points P_1, P_2, ...P_n. if number of points are not a multiple of 3 then point equivalent of space is padded with the message to make the number of characters a multiple of three. These points are the arranged in a 3xr matrix M. Another 3x3 non singular matrix A (such that |A| = ±1) is taken to compute Q=AM. The points on Q are then encrypted using any ECC encryption algorithm and sent over an insecure channel. In the receiving end after decryption receiver gets Q. Multiplying Q by A^{-1} she will get M.

Method – 3: This method [7] has taken into consideration only the alphanumeric (i.e. characters A-Z and digits 0-9) characters. Authors mapped these characters into the numbers from 0 to 35. A base value k (which is known to both sender and receiver) is chosen. For each character value n, x coordinate of mapped point P is obtained if an y on $E_p(a, b)$ is found for x = n.k. Otherwise x is incremented by 1 and again search for y. If not a y is obtained then x is again increased. Proceeding this way P(x, y) ∈ Ep (a, b) is obtained [8]. Then P is encrypted using ECC encryption and transmitted. Receiver after decryption obtains x value of P. she then computes $\lfloor x/k \rfloor$ to get corresponding character value.

III. PROPOSED MAPPING METHOD

In our proposed scheme instead of using ASCII values for each character we use a prime number to represent each character. Frequency analysis table from Oxford University Press, 2019 [9] have been followed. Following the above table 30 simple text message paragraphs have been studied. We have encoded each character in such a way that highest frequent character has the lowest prime value. The next higher frequent character has the next prime value. A table of 40 characters with assigned prime values is given below. Both the sender and receiver have to agree upon this table and it should be kept in a public file. For better performance here we consider only case insensitive alphanumeric characters with a few other characters which are generally used in only simple text messages.

Table-I

SL No	Character	Assigned Value
1	Space	2
2	E or e	3
3	A or a	5
4	R or r	7
5	I or i	11
6	O or o	13
7	T or t	17
8	N or n	19
9	S or s	23
10	L or l	29
11	C or c	31
12	U or u	37
13	D or d	41
14	P or p	43
15	M or m	47
16	H or h	53
17	G or g	59
18	B or b	61

19	F or f	67
20	Y or y	71
21	W or w	73
22	K or k	79
23	V or v	83
24	X or x	89
25	Z or z	97
26	J or j	101
27	Q or q	103
28	0	107
29	1	109
30	2	113
31	3	127
32	4	131
33	5	137
34	6	139
35	7	149
36	8	151
37	9	157
38	.	163
39	,	167
40	?	173

Using table-I each substring S (scanned from input string from left to right) of 9 characters will be encoded at a time. Each such S will produce two values N and x. We get N by multiplying character values of S taking repeated characters only once. This is done so that in receiving end factoring N, occurrences of each character can be obtained easily. Position values of characters from S are mapped in x. In receiving end from digits of x, S can be rearranged. N and x are now mapped on points of E_p (a, b). Thus each S will give two points on E_p (a, b).

The encoding algorithm is as shown below:
Input: A string S of 9 characters.
Output: Pair of points on E_p (a, b) as P(x1, y1) & Q(x2, y2).

1. Scan the input string from left to right taking 9 characters at a time. Call it as S.
2. Get S1 from S excluding repeated characters.
3. Sort S1 in ascending order according to values given in table - I.
4. Get a product N of character values in S1.
5. Form another 9 digit number x from S & S1 as follows:
    a. Set x = 0
    b. For i from 1 to 9
        i. Find position of i^{th} character of S in S1 call it y.
        ii. Set x= x*10+y.
6. Multiply N and x with some base value k known to both sender & receiver to get x1 & x2.
7. Use Koblitz's method [2] to find corresponding y1 & y2 on E_p (a, b).

In the receiving side after decryption we get points P & Q. From P & Q after decoding we get the original message. The decoding algorithm is as given below:

Input: Decrypted points P & Q on E_p (a, b).
Output: Original string of 9 characters.

1. From the decrypted points P(x1, y1) & Q(x2, y2) take x1 & x2.
2. Calculate x1 $\leftarrow \lfloor x1/10 \rfloor$ and x2 $\leftarrow \lfloor x2/10 \rfloor$.
3. Factorize x1.
4. Sort factors of x1 in ascending order and store them in S1.
5. Set string S = null.
6. Execute following steps 9 times.
    a. Extract next digit i from x2 (start from LSB).
    b. Find i^{th} element from S1. Say it as j.
    c. From table 1 take character c whose value is j.
    d. Append c in S.
7. Reverse S to get original substring.

Examples and illustration: In the sender side consider the string "A text message mapping algorithm using points on elliptic curve". Taking the first 9 character substring i.e. S = "A text me" we get the product of character values as N = 2x3x5x17x47x89 = 2133330 where Sorted S1 = {2, 3, 5, 17, 47, 89}. Also we get x as 314264152. Taking the base value k as 10 we get x1 = 21333300 and x2 = 3142641520. Let us take the NIST recommended curve [10] over a prime field GF (p) as

$$y^2 = x^3 + ax + b \pmod p \qquad (1)$$

Where a = -3,
b=245515554600894381774029391519745178476910805816119123806 5
p=627710173538668076383578942320766641608390870039032496127 9 to find out
y1=333749624378296539152901406503880377130683252461562857308;
y2=273257579981523815456912667042789227489738105993852890377 3;
The points P & Q obtained in this way are thus:
P = (21333300,
333749624378296539152901406503880377130683252461562857308);
Q = (3142641522,
273257579981523815456912667042789227489738105993852890377 3);

These points can now be transmitted encrypting P & Q using elliptic curve analogue of the Massey-Omura system or elliptic curve analogue of ElGamal system. In the receiving end after decryption again we get the points P & Q from which we decode the original message as follows:

From points P & Q take x1 as 21333300 and x2 as 3142641522. First calculate x1 = $\lfloor x1/10 \rfloor$ = 2133330 and x2 = $\lfloor x2/10 \rfloor$ = 314264152. Factorizing x1 we get factors as the sequence F= {2, 3, 5, 17, 47, 89} from which consulting table-I we get the characters as ' ', 'e', 'a','t', 'm', 'x'. For convenience we take only lowercase characters from two alternatives for each value specified in table-I. Now extract rightmost digit from x2 as 2. This value indicates the position of last character in F. Since in F 2nd value is 3 so the last character is e. Similarly we obtain the last but one character's position is 5 from x2. Value of 5th element in F is 47. So the last but one character is m. In this way we get a string as "em txet a" reversing which we get "a text me".

IV. RESULTS

In the table-II given below we show number of bits required to encode the message "A text message mapping algorithm using points on elliptic curve" according to the above mentioned schemes using the curve in (1). All calculations are done using magma [11]. The base point G(x, y) is taken from [10] as follows:

x=602046282375688656758213480587526111916698976636884684818

y=174050332293622031404857552280219410364023488927386650641

Table-II

Message mapping schemes	No of bits in encoded points
Method-1	24104
Method -2 (Taking 3 x 3 matrix from [6])	24075
Method -3 (value of space is taken as 36)	22538
Proposed Method	3129

From the above result it is clear that the number of bits used in the proposed method is reduced remarkably. Also group of several characters used in a single point, thus make attacks, analyzing frequency of characters, impractical. So it is a secure method.

V. CONCLUSIONS & FUTURE WORK

In this paper a new method of converting plaintext messages into Elliptic Curve points is proposed. The method is implemented for a set of 40 characters that are generally used in transferring simple plaintext messages. But it can be extended for the full set of 7 bit ASCII characters. Grouping of 9 characters is taken for efficiency i.e. to reduce bits to be transmitted. With a little bit degraded performance more than 9 characters grouping is also possible. A rigorous checking is necessary before practically using it.

REFERENCES

[1] Victor Miller, "Uses of Elliptic Curve in Cryptography", Advances in cryptology-CRYPTO'85, vol-218, SpringerHeidelberg, 1986, pp. 417-426.

[2] Neal Koblitz, "Elliptic Curve Cryptosystems", Mathematics of Computation, Vol-48, 1987, pp-203-209

[3] Alfred J Menezes, Paul C. van Oorschot, Scott A Vanstone, "A Handbook of applied Cryptography". CRC Press.

[4] Darrel Hankerson, Alfred Menezes, Scott Vanstone "Guide to Elliptic Curve Cryptography", Springer Professional Computing, 2004.

[5] William Stallings, "Network Security Essentials: Applications and Standards", 4th Edition.

[6] F.Amounas and E.H.EI Kinani, "Fast mapping method based on matrix approach for Elliptic Curve cryptography", International Journal of Information and Network Security 1, 54-59(2012).

[7] Bh P, Chandravati D, Prapoorna Roja P, "Encoding and decoding of a message in the implementation of Elliptic Curve Cryptography using Koblitz's method." International Journal on Computer Science and Engineering, 2010; 2(5): 1904-1907.

[8] Aritro Sengupta and Utpal Kumar Ray, "Message mapping and reverse mapping in elliptic curve cryptosystem", Security Comm. Networks 2016;9:5363:5375.

[9] Oxford University Press, 2019, "which-letters-are-used-most".

[10] Recommended elliptic curves for federal government use; July 1999.

[11] John Cannon, Wieb Bosma, Claus Fieker, Allan Steel (Editors), "Handbook of Magma Functions", Version 2.19.

Authors Profile

Mr T. K. Ghosh, MCA (BU, 2002) has been working as an Assistant Professor in Department of Computer Science, Bankura Sammilani College, (a.t. Bankura University), PO: Kenduadihi, DT: Bankura, WB, India, since 2008. His primary thrust area is Cryptography.

