|| Volume 3 || Issue 11 || November 2018 || ISSN (Online) 2456-0774

INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

Secure Method for Text Encryption using Elliptic Curve Cryptography

Ms. Gayke Geeta Ganesh

P.G Student, Department of Computer Science and Engineering, CSMSS Chh. Shahu College of Engineering, Aurangabad, Dr. BAMU, Maharashtra, India.
geetgayke@gmail.com

Abstract:- This paper presents a secure policy for text Encryption by means of Elliptic Curve Cryptography (ECC) where Cryptography is a transformation of unadorned message to make them protected and Insusceptible from intruders. Hidden data from Cryptography can only be viewed by the authorized people. Elliptic curve cryptography has been a modern study area in the field of cryptography. In other words, Elliptic curve cryptography is a methodology to public-key cryptography based on algebraic structure of elliptic arch (curve) over predetermined fields.

A modern method has been projected in this research paper where typical system of plotting character to affine points in elliptic arc has been removed. This technique will eliminate the expensive process of mapping whereas it pop ups an essential need to share the common lookup chart amongst the sender and the receiver. The algorithm is planned in such a mode that it can be used to encrypt or decrypt, several forms of script with definite ASCII values. Here, the procedure of Encryption and decryption is implemented very smoothly. Even with abundant words as input this technique serves, smaller size of cipher text as compared to previous techniques. As ECC equally, provides security & resolves the problem of large key size, it is appropriate for campaigns which have power, storage and processing limitation.

Keywords- *Cryptography, Elliptic curve cryptography, Cipher text , Encryption and decryption.*

I INTRODUCTION

Elliptic curve cryptography was introduced in 1985 by Neal Kobilt & Victor Miller. Cryptography involves the conversion of ordinary plain text into secure codes which allow information to be kept secret and it is derived from mathematical concepts and a set of rule-based calculations called algorithms. The chief portion of cryptography is Encryption and Decryption. Common cryptographic primitives are RSA, DSA hash functions.

For example 224 bit Elliptic curve provide same security as that of a 2048 bit RSA. ECC generates keys through the properties of the elliptic curve equation such as:

$$y^2=x^3+ax+b$$

The elliptic curvature or arch over major field encompasses the constants of the elliptic curvature or arch & the base point, which is a Point situated on the arch. The arch or curve designated exposed be recognised to both dispatcher and the receiver .The main action in ECC is scalar multiplication which includes of Point addition and Point doubling.

The Encryption is completed by converting the message note into poles (points) on the curvature and then performs scalar multiplication using dispatcher's private key. The poles (points) are translated into the genuine message after decryption. The charm of using elliptic curves rises from the fact that analogous level of security can be achieved with considerably shorter keys than in methods based on the difficulties of solving discrete logarithms over integers or integer factorizations.

In this research we have introduced text Encryption and Decryption techniques that transform the message into to affine points on the curve. Encryption techniques change the plain ordinary text into ASCII standards and then transform this into BIG INTEGER. The converted BIG INTEGER values are clustered together to generate the x and y poles. The transformed standards are encrypted in contrary order to form original message. The contrary order encryption offers more safety for the text encryption.

II PROBLEM STATEMENT

Numerous scholars who have applied text encoding and decoding by the use of ECC, this cryptography did agreement for the table which contain characters ,ECC poles or coordinates mapping ,by using ASCII standards of character to originate affine elliptic curvature poles on the performance of multiplication action of point with 'G' as generator & its equivalent ASCII standard for the character. A novel idea is introduced here that the usage of plotting on look up table commonly amongst the dispatcher and receiver can be entirely removed. This report is planned to encode on blocks consisting numerous characters which results that the methodology is also appropriate for lengthy size of data. There is no limitation of English script on algorithm, so procedure could be used for any given script which has predefined ASCII values.

Objectives

1) This scheme offers advanced level of safety with smaller key size related to previous cryptographic techniques.

2) The further objective is the message is encoded by means of private key of dispatcher and decoding is done by means of dispatcher's public key as well as receiver's private key.

3) System escapes the expensive action of mapping or plotting, & also escapes the need of sharing common look up table between both the ends.

III LITERATURE SURVEY

In literature survey some of the approaches are underlined by us, whose resultant, is the security of encryption method in elliptic curve cryptography . In 1987, public key crypto-system elliptic curve over finite field is introduced by Neal Koblitz [2] multiplicative group were used by them for more security. They also state that elliptic curve is harder on discrete logarithmic problem, so now it is safer compared to additional public key crypto-systems. The extended idea of Neal

Koblitz, Alfred Menezes and Scott Vanstone [3] in 2000, was the usage of public key cryptography by Diffie-Hellman, to elliptic curve group on discrete logarithmic problem, which provides smaller block size, great speed and great security. A Book is written by Darrel Hankerson, Alfred Menezes and Scott Vanstone[4] in 2004 named as Guide to Elliptic Curve Cryptography which gives several facts of elliptic curve arithmetic, cryptographic protocols and operational issues. On firstly transforming the message in ASCII values, with the mapping of affine points on Elliptic curve, by the use of point addition of the ASCII value time Generator & ECC, by S. Maria Celestin and K. Muneeswaran [7] in 2009, given rise to implemented text cryptography technique . In terms of mathematical implementation Jorko Teeriaho [6] in 2011 gave example of elliptic curve key exchange, encryption and digital signature.

IV PROPOSED SYSTEM

The proposed system is an Elliptic curve cryptographic system where the message encryption using private key of dispatcher and depiction is done using public key of dispatchers as well as private key of receiver. For performing Elliptic curve encryption and decryption both the ends uses following Elliptic curve equation:

$$y^2=x^3 + ax + b \bmod p$$

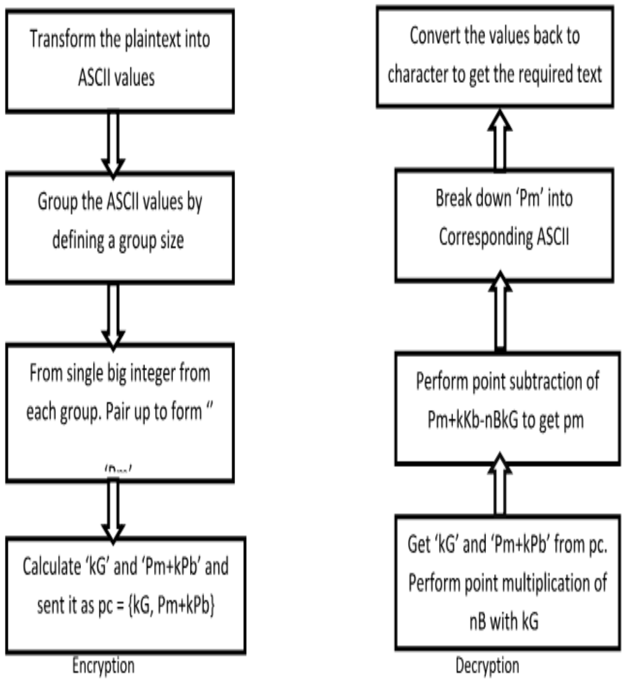
Transform the plaintext into ASCII values
Group the ASCII values by defining a group size
From single big integer from each group. Pair up to form 'Pm'
Calculate 'kG' and 'Pm+kPb' and sent it as pc = {kG, Pm+kPb}
Encryption
Convert the values back to character to get the required text
Break down 'Pm' into Corresponding ASCII
Perform point subtraction of Pm+kKb-nBkG to get pm
Get 'kG' and 'Pm+kPb' from pc. Perform point multiplication of nB with kG
Decryption

Figure 1 Block diagram for encryption/decryption process.

In Encryption process the Plaintext is converted into the ASCII value, then Partition the ASCII values by defining a group size where the Group size is given by:

groupsize = Length[IntegerDigits[p, 65536]]− 1

This action assembles the ASCII values, with the size given by group, without overlapping & later they left as sub list , that is smaller in size compared to group size, as it is without padding. By calculating group the ASCII values are transformed into big integer value taking base as 65536 and pad with 32 to the end of list by the group size. A single big integer we get 'P_m' as point message. As 32 represent the blank space in ASCII code, we pad with 32. 'k' value is randomly selected with generator 'G' where of 'k' ranges from 1 to n-1. Calculate kG and kpb by the use point multiplication. Where P be any point on the elliptic curve, in point multiplication operation. Multiplication operation over P is well-defined by repeated action of addition.

By using point addition and point doubling, required calculation of Pm + kPb are made.

P and Q be the two points on the curve for point addition, P+Q = R (x3,y3) is the result of adding P and Q, the following calculation is given by

$$kP = P + P + P +\cdot\ \cdot\ \cdot+k \text{ times.}$$

$$x_3 = \{\lambda^2 - x_1 - x_2\} \bmod p$$

$$y_3 = \{c\,(x_1 - x_3) - y_1\} \bmod p$$

Where,

$$\lambda = y_2 \_ y_1 \ / \ x_2 \_ x_1$$

In point doubling operation P be a Point on the curve, Doubling results in R=2P.The computation is pictorially given by drawing tangent on the point P. The two point P(x_1, y_1) and Q(x_1, y_1) overlap. P + Q = R(x_3,y_3) is given by the following calculation.

$$x_3 = \{\ \lambda^2 - 2x_1\ \} \bmod p$$

$$y_3 = \{\ \lambda(x_2 - x_1) - y_1\ \} \bmod p$$

Where,

$$\lambda = 3x_1^2 + a \ / \ 2y_1 \ \bmod p$$

Send cipher text P_c = {kG, Pm + kPb} to the receiver side.

A reverse process of encryption is called as Decryption. It is process of converting Cipher text into Plain text. Cryptography uses the decryption technique at the receiver side to obtain the original message from non-readable message (Cipher Text). The process of decryption requires two things such as Decryption algorithm and a key. A Decryption algorithm means the technique that has been used in Decryption. Now get cipher text P_c from sender side. Separate left part kG and right part Pm + kPb from the ciphertext P_c. perform point multiplication operation of nB with kG to the left part and subtract it from the right part to get P_m.

$$\{Pm + kPb\} - nBkG = Pm$$

Subtraction operation can be converted to addition by multiplying with -1 to the y coordinate. Converting- 'Pm' into corresponding ASCII values. And convert the ASCII values back to characters to get the required text.

Encryption Algorithm:

1. Let p be Plaintext
2. Change p to ASCII values
3. Set groupsize = Length[IntegerDigits[p,65535]] – 1
   Where p = given Integer i.e. ASCII values of Integer
4. Partition ASCII value into groups
   Partition [ASCII values, group size, groupsize, 1,{}]
5. Convert each group to form digits with base 65535
   FromDigits [Group of ASCII values, 65536]

6. Pad list with 32 at the end
   Pad count list with 32 while count list is not empty
   If count list == odd
   Pm=countlist + 32
7. Select K as random from 1 to n-1
   // compute k with G and P_b with point multiplication
8. Compute c = P_m + KP_b
   // using point addition or point doubling
9. P_c = (KG, P_m + KP_b)
   // Where P_c= {kG, Pm + kPb}

Decryption Algorithm:

1 Get the cipher text P_c

2 Let kG be the first point and Pm + kPb be the second point

3 nBkG = nB * first point;

4 Calculate Pm1 = Pm + kPB- nBkG;
// Where Pb = nBG.

5 Calculate the Pm value from Pm1 using discrete logarithm

Encryption Example:

- Input text. the text is :
  hello world
- Its equivalent ASCII values are :
  {104, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100}
- Group the ASCII values with size calculated as Length [IntegerDigits[p,65536]]-1 which we get as 3 {{104,101,108,108}{111,32,119,111}{114,108,100}}
- Covert each group into big integers using FromDigits function with base 65536.
  {29273831376683116,31243859861635183, 489633349732}
- Pad with 32 at the end of the above list if the number of term is odd, so that paring can be done.
  {29273831376683116, 31243859861635183, 489633349732,32}
- After calculating cipher text, Pc = {kG, Pm+kPb} is obtained as:

kG=
(602046282375688681773291964883006405209242039197525606 40.000,17405033229362203399543947570408672204940937760063 0947840.000)

kPb=
(280300078654161747795749502030115792226924309684027601 7152.000,42697180211059444468581135308995368056806746769 99297433600.000)
Pm+kPb=
(29273831376683116.000,31243859861635184.000,489633349732000, 32.000)

P_c = {kG, P_m+ kPb}

- Send the cipher text Pc to the communicating party.

Decryption Example:

- Obtain the cipher text Pc i.e. kG and Pm + kPb.

kG=(60204628237568868177329196488300640520924203919 75256064.000,1740503322936220339954394757040867220494 09377600630947840.000)
Pm + kPb =
(29273831376683116.000, 31243859861635184.000)

- Perform Point multiplication using the private key of the sender nB to kG :
  nBkG =
  (602046282375688681773291964883006405209242039197525 60640.000,174050332293622033995439475704086722049409 377600630947840.000)
- Perform point addition operation with the above result with Pm+kPb :
  {29273831376683116, 31243859861635183, 489633349732, 32}
- Convert the above result to ASCII values using IntegerDigits function with base 65536 :
  {104, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100, 32}
- Perform conversion operation of ASCII values to characters get the required message:
  hello world

V FLOW DIAGRAMS

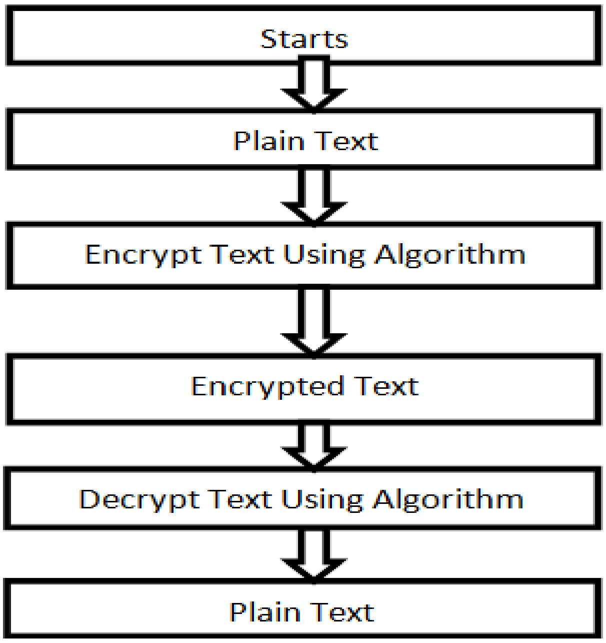

Figure 2 Flow Diagram

Initially text is allotted to input text and this text is transforming into ASCII values then dividing the ASCII values i.e. this action assembles the ASCII values, with the size given by group, without overlapping and converting ASCII values into BIG INTEGER taking base as 65536. Pad with 32 to the end of the list from the above step if the count of the above list is odd, to make it even for performing complete pairing. Then by means of the action of point addition or point doubling and point multiplication cipher text is calculated. Algorithm 1 explain the Encryption process of the System.

VI TIME EFFICIENCY OF THE SYSTEM

Performance Comparison:

Method	No. of Words	Encryption Time(second)	Decryption Time(second)
Our Method	5	0.0015154039	0.0021922039
Implementation of Text Encryption using Elliptic Curve Cryptography	5	0.09	0.10
Implementation of Text based cryptosystem using elliptic curve cryptography	5	1.95	0.83

Time Required to Encryption

The following figure 3 represents the output result of the Encryption Time for the given sized message. The time of the System is measure in Seconds. So that Encryption time of the system for 5 words is 0.0006725745356277342 seconds, 3 words is 0.0006544,9 words is 0.000905620, 20 word is 0.00115859 and 30 words is 0.001332469.

Time Required to Decryption

The above figure 4 represents the output result of the Decryption Time for the given sized message. Decryption Time of the System for 5 words is 0.002031609795693612 seconds, 9 words is 0.00322159,13 words is 0.00332302,20 words is 0.00503887, 30words is 0.0064347352.

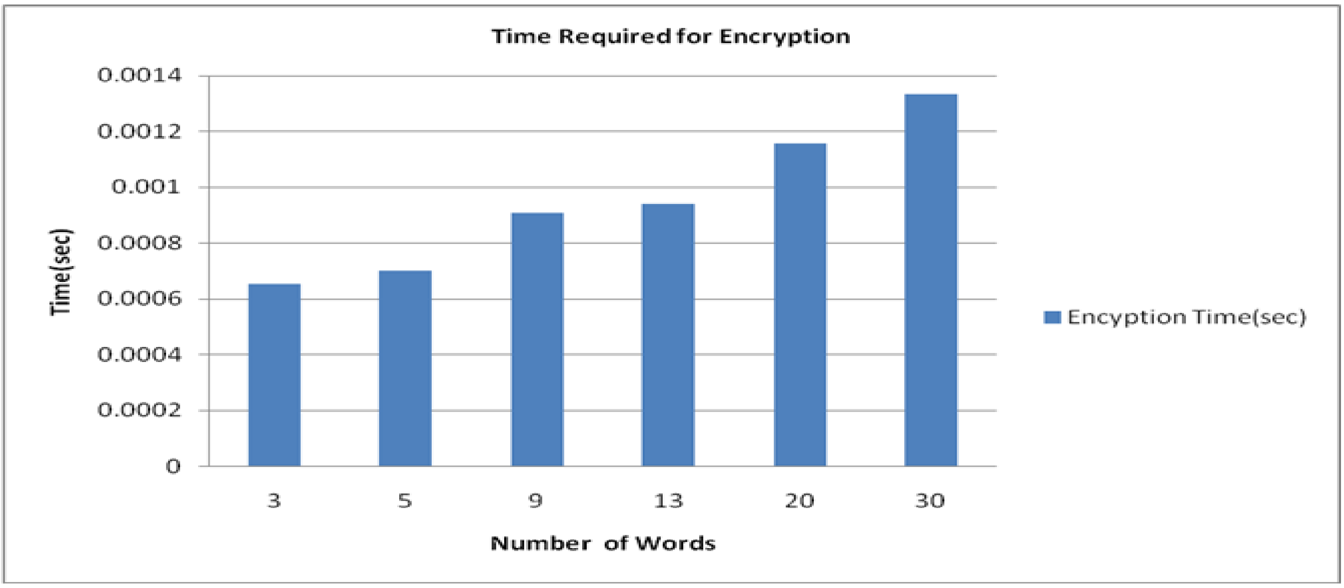

Figure 3 Time Required to Encryption

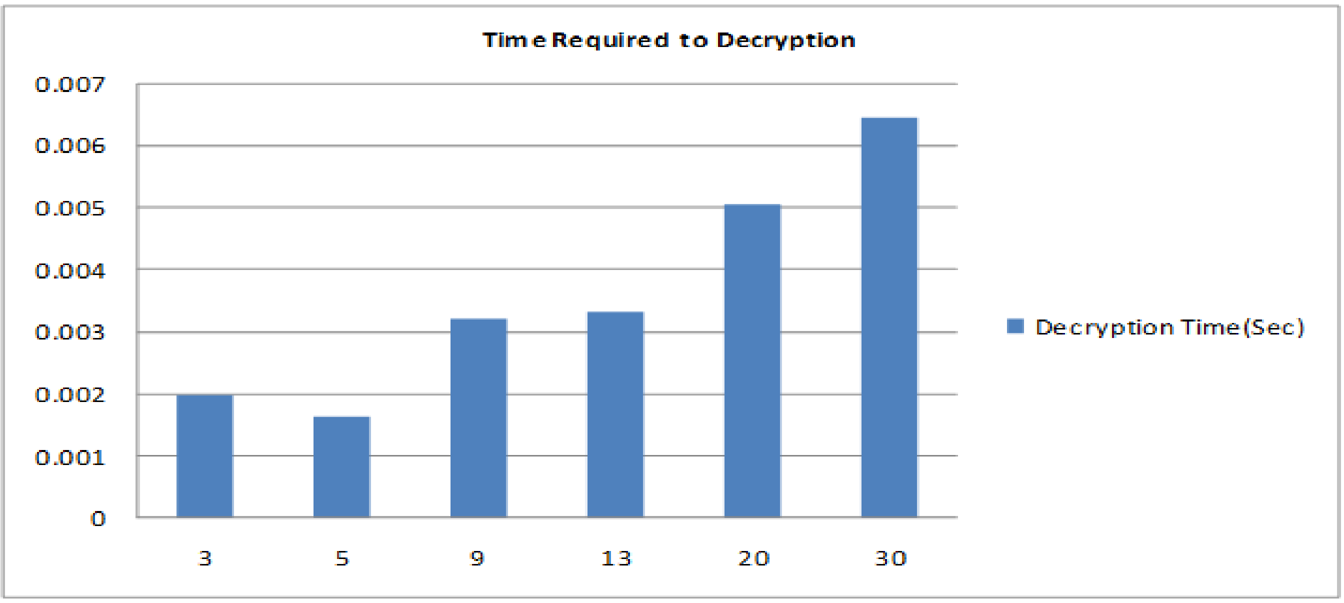

Figure 4Time Required to Decryption

VII CONCLUSION

In this research, the methodology for text encryption using elliptic curve cryptography is found as secured which the aim of this paper is. This novel technique is intended deliver smaller size cipher text related to supplementary cryptographic method while implementing encoding and decoding process smoothly with input as abundant words. This theme is designed to encrypt on blocks consisting numerous characters which results that the methodology is also suitable for large size data. The further improvement is the message is encoded by means of private key of dispatcher and decoding is done by means of dispatcher's public key as well as receiver's private key. System sidesteps the expensive action of mapping or plotting, & also escapes the need of sharing common look up table between both the ends.

REFERENCES

[1] Victor S. Miller, Use of Elliptic Curves in Cryptography, Advances in Cryptology-CRYPTO'85 Proceedings, Springer, vol. 218, pp. 417–426,December (2000).

[2] Neal Koblitz, Elliptic Curve Cryptosystems, Mathematics of Computation, vol. 48, issue 177, pp. 203–209, January (1987).

[3] Neal Koblitz, Alfred Menezes and Scott Vanstone, The State of Elliptic Curve Cryptography, Designs, Codes and Cryptography, vol. 19, issue 2–3, pp. 173–193, (2000).

[4] Darrel Hankerson, Alfred Menezes and Scott Vanstone, Guide to Elliptic Curve Cryptography, Springer (2004).

[5] Lawrence C. Washington, Elliptic Curves Number Theory and Cryptography, Taylor & Francis Group, Second Edition (2008).

[6] Jorko Teeriaho, Cyclic Group Cryptography with Elliptic Curves, Brasov, May (2011).

[7] S.Maria Celestin Vigila and K. Muneeswaran, Implementation of Text based Cryptosystem using Elliptic Curve Cryptography, International Conference on Advanced Computing, IEEE, pp. 82–85, December (2009).