International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

IMPLEMENTATION OF ELLIPTIC CURVE CRYPTOGRAPHY ON TEXT AND IMAGE

Mrs. Megha Kolhekar
Assistant Professor, Department of Electronics and Telecommunication Engineering
Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai - 400703, India

Mrs. Anita Jadhav
Lecturer, Department of Electronics and Telecommunication Engineering
Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai - 400703, India

ABSTRACT- In recent years, Elliptic Curve Cryptography (ECC) has attracted the attention of researchers and product developers due to its robust mathematical structure and highest security compared to other existing algorithms like RSA (Rivest Adleman and Shameer Public key Algorithm). It is found to give an increased security compared to RSA for the same key-size or same security as RSA with less key size [1]. In this paper, we give a brief background of key exchange and encryption/decryption using ECC. We then explain implementation of these algorithms on text documents. We have used C++ as the tool for implementation.

Index terms- ECC, Image encryption, Key Exchange, Text encryption

INTRODUCTION

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 a secret.

The idea of using Elliptic curves in cryptography was introduced by Victor Miller and N. Koblitz as an alternative to established public-key systems such as DSA and RSA. The Elliptical curve Discrete Log Problem (ECDLP) makes it difficult to break an ECC as compared to RSA and DSA where the problems of factorization or the discrete log problem can be solved in sub-exponential time. This means that significantly smaller parameters can be used in ECC than in other competitive systems such as RSA and DSA. This helps in having smaller key size hence faster computations.[1]

We have studied application of Elliptic Curves over finite fields 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. We describe this furthur in this paper.

ELLIPTIC CURVES OVER FINITE FIELDS

An elliptic curve $E(F_p)$ over a finite field F_p is defined by the parameters a, b $\in$ F_p (a, b satisfy the relation $4a^3 + 27b^2 \neq 0$), consists of the set of points (x, y) $\in$ F_p, satisfying the equation $y^2 = x^3 + ax + b$. The set of points on $E(F_p)$ also include point ***O***, which is the point at infinity and which is the identity element under addition.

Figure 1 Addition of 2 points P and Q on the curve $y^2 = x^3 - 3x + 3$

The Addition operator is defined over $E(F_p)$ and it can be seen that $E(F_p)$ forms an abelian group under addition.

The addition operation in $E(F_p)$ is specified as follows:

- $P + \boldsymbol{O} = \boldsymbol{O} + P = P, \forall P \in E(F_p)$

- If P = (x , y) $\in$ E(F_p), then (x, y) + (x, – y) = ***O***. The point (x, – y) $\in$ E(F_p) and is called the negative of P and is denoted as –P)
  - If P = (x_1, y_1) $\in$ E(F_p) and Q = (x_2, y_2) $\in$ E(F_p) and P $\neq$ Q, then R = P + Q = (x_3, y_3) $\in$ E(F_p), where $x_3 = \lambda^2 - x_1 - x_2$,

$$y_3 = \lambda (x_1 - x_3) - y_1, \text{ and } \lambda = (y_2 - y_1) / (x_2 - x_1)$$

i.e. the sum of two points can be visualized as the point of intersection of E(F_p) and the straight line passing through both the points.

Let P = (x, y) $\in$ E(F_p). Then the point Q = P + P = 2P = (x_1, y_1) $\in$ E(F_p), where $x_1 = \lambda^2 - 2x$, $y_1 = \lambda (x - x_1) - y$, and $\lambda = (3x^2 + a) / 2y$. This operation is also called doubling of a point and can be visualized as the point of intersection of the elliptic curve and the tangent at P.
The reason for choosing prime fields is that **distinct additive and multiplicative inverses** exist for each number i.e. 0 to (P-1) in the field of the prime number P.

Figure 2 Doubling of a point P, R = 2P on the curve $y^2 = x^3 - 3x + 3$

ELLIPTICAL CURVE DISCRETE LOGARITHM PROBLEM (ECDLP)

The strength of the Elliptic Curve Cryptography lies in the Elliptic Curve Discrete Log Problem (ECDLP). The statement of ECDLP is as follows:

Let E be an elliptic curve and P $\in$ E be a point of order n. Given a point Q $\in$ E with Q = mP, for a certain m $\in$ {2, 3, ……, m – 2}.

Find the m for which the above equation holds.

When E and P are properly chosen, the ECDLP is thought to be infeasible. Note that m = 0, 1 and m – 1, Q takes the values ***O***, P and – P. One of the conditions is that the order of P i.e. n should be large so that it is infeasible to check all the possibilities of m by brute force attack.

E.g. For P≡(2,2) and Q≡(153,108) such that Q=5P,then the discrete logarithm of Q to the base P is 5.

ECDLP problem: Given the values of P and Q ,it is difficult to find the scalar multiple n.

While Discrete Logarithm Problem(DLP) is defined as: **$b \equiv \alpha^a \pmod p$,** where, 'a' is the discrete logarithm of b to the base α mod p. Given the value of index'a',it is easy to find discrete logarithm 'b'. But given the value of 'b',it is difficult to find the index 'a'. For example
$(2^{13}) \bmod 19 = 3$, i.e. discrete log of 3 for the base 2 mod 19 is 13.

The difference between ECDLP and the Discrete Logarithm :
Problem (DLP) is that, DLP though a hard problem is known to have a sub exponential time solution, and the solution of the DLP can be computed faster than that to the ECDLP. This property of Elliptic curves makes it favorable for its use in cryptography.

ECC KEY EXCHANGE

Figure 3 Key Exchange in ECC

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 need 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.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

ELLIPTIC CURVE ENCRYPTION/DECRYPTION

1. Consider a message 'Pm' sent from A to B. 'A' chooses a random positive integer 'k' , a private key 'n_A' and generates the public key $P_A = n_A \times G$ 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

   $$Pm + kP_B - n_B(kG) = Pm + k(n_B G) – n_B(kG)=Pm$$

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:-

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

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

X	Y
167	30
167	181
179	12
191	15

Table 1 Elliptic curve points

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.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

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 :-

i. Mapping-1 is used to convert an integer into a corresponding elliptic curve points from our database.
ii. Mapping-2 is used to convert an elliptic curve point into its corresponding integer from our database.
iii. Reverse Mapping-1 does the same function as Mapping-2.
iv. Reverse Mapping-2 does same function as Mapping-1.

These codes perform scanning operations.

For example:

Database	Curve Point	
Point No.	X	Y
15	16	100
16	16	111
17	17	30
18	17	181
19	19	37
20	19	174
21	20	20
22	20	191
23	21	87
24	21	124

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

Table 2 Database with integers and corresponding elliptic curve points

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.

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. Printable ASCII character ($) which acts 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.

Figure 4 Text Encryption Flow chart

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.

Figure 5 Text Decryption Flow chart

FEATURES OF THE TEXT 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 substitutional ciphering technique.

This problem is solved by using **many-to-one mapping** for characters and the original characters are differentiated 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 that 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^DEEE~+G

Decrypted text

This is document number 2 for encryption.

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.

Figure 6 Image Encryption Flowchart

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.

Figure 7 Image Decryption Flowchart

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

CONCLUSION

In this paper, we have given a brief description of ECC key exchange and encryption/decryption. We have explained in detail the implementation of ECC on text document in C++. Our scheme of encryption is simple, exploits all security features of elliptic curves and is applicable to all ASCII characters. Since

we have not come across any reference that gives explains scheme of encryption of text using ECC, (although there are many references explaining ECC), we may claim that the scheme described in this paper is our contribution.

ACKNOWLEDGEMENTS

We are thankful to our former students (2006-2010), Mr. Anup John Joseph, Mr. Mandar Koli, Mr. Kamal Kishor Pal, and Mr. Neeraj Savant for carrying out the software implementation of the algorithms.

REFERENCES

[1] William Stallings, Cryptography and network security, 2nd edition, Prentice Hall publications

[2] B.Schiener, Applied Cryptography. John Wiley publications and sons, 2nd edition, 1996

[3] Victor Miller, “Uses of elliptic curves in cryptography”, Advances in cryptology, 1986

[4] N.Koblitz, A course in number theory and cryptography.

[5] www.prime-numbers.org

[6] www.rsa.com/rsalabs