Volume 4, Issue 1, January 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Encryption using Elliptic Curve Cryptography using Java as Implementation tool

Ikshwansu Nautiyal*
Dept. of Information Technology
DIT University, Dehradun, India

Madhu Sharma
Asst. Prof., Dept. of Computer Science
DIT University, Dehradun, India

Abstract— Cryptography is the technique of hiding a message in some unintelligible format so that the message lies hidden in plain sight of an unintended person. The techniques of cryptography are centuries old. With technological advancement, techniques have evolved significantly. Public key cryptography offers a wide range of security over the various modes of transferring data, especially over Internet. The security of a public key encryption is stronger only if the authenticity of the public key is ensured. Data encryption standards like RSA and Diffie- Hellman are becoming incapable due to requirement of large number of bits for cryptographic process. As of latest, ECC (Elliptic Curve Cryptography) has become the latest trend in the cryptographic scenario. This paper presents the implementation of ECC for encryption/decryption and authentication process, using JAVA as the implementation tool. It is worth noting that brute force attack on ECC is infeasible due to the discrete logarithm problem it possesses.

Keywords— Elliptic Curve Cryptography (ECC), Discrete Logarithm, Elliptic Curve (EC), public-key cryptography.

I. INTRODUCTION

Neil Koblitz and Victor Miller first proposed the use of Elliptic Curves in Cryptography in 1985. A lot of work has been done on ECC since then. Elliptic curves fascinates a lot of cryptographic work due to the fact that despite the use of smaller keys they provide same level of security than RSA or key exchange algorithm like Diffie-Hellman. ECC makes use of elliptic curve defined over a finite field. A finite field restricts the variable and coefficients to its elements. Elliptic Curves are not ellipses, and are not to be confused with them. The only feature that elliptic curves share with an ellipse is the nature of equation that generates the elliptic curve. For cryptographic processes it is necessary that the elliptic curves be defined over a finite field, typically a prime finite field, so that the decryption process be carried out within the range of the elements. Otherwise, it will be impossible to apply the cryptographic process.

In ECC, we start with a Base Point (B_P) and an affine point A_P (x, y). A Base Point is the smallest co-ordinate on the elliptic curve, satisfying the elliptic curve equation. An affine point AP may or may not be the Base Point, though it must be close to the base point, in latter case. For the encryption process, first a character is transformed into its ASCII code and then with a scalar multiplication with the pre selected affine point (A_P) is converted into another affine point on the EC. That is, lets say we have the ASCII value of a character as 'p', then we determine A_{PL}= p*A_P. The newly calculated A_{PL} is another affine point obtained, that also lies on EC. The multiplication is obtained by applying repeated addition strategies of ECC technique. Then as per ECC algorithm, we add the newly obtained point A_{PL} with kP_{UB}, where k is a randomly generated large secret integer and P_{UB} is the public key of receiver (User B). The addition yields us to another affine point (A_{PL}+ kP_{UB}) of the EC. This is the second part of the encrypted message to be sent. The first part constitutes the product of secret integer k and Base Point G, i.e. kB_P.

Thus the encrypted message to be sent to the receiver is (kB_P, A_{PL}+kP_{UB}). These are two sets of co-ordinates on the EC.

For the decryption part, first we apply receivers private key, P_B, on the first part, i.e., kB_P. This yields us kP_{UB}. This is then subtracted from the second part of the received message, which provides us with the A_{PL}.

P_B*kB_P = kP_{UB}

(A_{PL}+kP_{UB}) - kP_{UB} = A_{PL}

Once we receive the A_{PL}, discrete logarithm concept can be applied to retrieve the ASCII value stored in the affine point. Hence the keys are transformed over the EC field for both encryption and decryption. This promises to afford maximum security from intruders and hackers.

II. RELATED WORKS

Elliptic Curve Cryptography has been thoroughly studied for the past two decades. Certicom, a BlackBerry subsidiary, in the field of Elliptic Curve Cryptography, pursues a significant work. The application of ECC and practical implementation crypto system in real world constitutes interdisciplinary research in computer science engineering. A hardware implementation involves research in the electrical and electronic engineering. ECC provides an excellent solution of data and secure transfer of keys between two communicating parties. S. Maria Celestin Vigila shows the basic implementation of ECC for cryptographic purpose [1]. An ECC based communication system, where the various characters can be represented as the co-ordinates of the EC, can also be generated [2]. A novel idea of ECC encryption and decryption based on Knapsack also shows on how the EC technology can be used for encryption and decryption [3].

Using Koblitz method one can also achieve the cryptographic process [4]. The work done by Vivek Kapoor shows use of Elliptic Curve Cryptography for smart cards [6]. Study of software implementation of ECC over Binary fields on workstations of the NIST-recommended elliptic curves and its implementation in C on a Pentium II 400 MHz workstation is shown in [7]. An implementation of elliptic curve cryptography over F_{2p} for sensor networks based on the 8-bit, 7.3828-MHz MICA2 mote has also been identified [8]. M.Aydos et.al [9] has presented an implementation of ECC over the field GF(p) on an 80 MHz, 32 bit RAM microprocessor along with the results. Kristin Lauter as provided an overview of ECC for wireless security [10].

III. PROPOSED METHOD DESCRIPTION

A general Elliptic Curve equation takes the general form:

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

(x, y) = co-ordinates on the EC.

a, b = coefficients

However for finite fields a modified equation is used [5]:

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

where p = prime number for which the EC be defined

a, b satisfy the equation [2]

$$(4a^3 + 27b^3) \bmod p \neq 0 \mod p \qquad (3)$$

An elliptic curve E over GF (p) consist of the solutions (x, y) defined by (1) and (2), along with an additional element called 0, which is the point of EC at infinity. The set of points (x, y) are said to be affine coordinate point representation.

The basic operations on elliptic curves are addition and doubling. A scalar multiplication with a point can be represented as a combination of addition operations.

Say, given a point P (x, y) is to be multiplied k times, say k = 37. Thus we have to calculate 37P.

In terms of addition it can be represented as

37P = P + P + P +… + P (37 times)

For addition of two points P(x1, y1) & Q(x2, y2). First calculate the tangent to the curve at point P [5].

$$L = [(y2 - y1)/(x2 - x1)] \bmod p, \text{ for } x1 \neq x2 \qquad (4)$$

$$L = [(3x1^2 + a)/ 2y_P] \bmod p, \text{ for } x1 = x2 \qquad (5)$$

P+Q = R

$$x3 = (L^2 - x1 - x2) \bmod p$$

$$y3 = (L (x1 - x3) - y2] \bmod p$$

where

Q = (x2, y2) = co-ordinates of Q change with every addition method applied, i.e. after first addition its co-ordinates will be (x3, y3), and so on.

L = tangent to the curve at point P.

R = (x3, y3), resultant co-ordinates

Thus applying addition methods, determined by the value of 'k', we obtain point R (x3, y3). Each point thus obtained is also an affine point on the Elliptic Curve.

IV. PROPOSED ALGORITHM

First of all, the points are generated for the elliptic curve based on the values of prime modulo p and predefined coefficients a, b (It is to be noted that 'a' and 'b' remain constant throughout the application of an elliptic curve).

GenPoints (prime, a, b)

```
{
Step 1:   initialize x = 0;
Step 2:   while (x<p)
          y^2 = (x^3 + ax + b) mod prime;
          if (LHS = RHS)
          output (sqrt (x), sqrt (y));
          x = x+1;
}
```

Algorithm ECC

```
{
```

Algorithm for Key Distribution

Step 1: //For user A

$P_{UB} = G*P$

$U_A = (P_{UA}, P_A)$ // User A key pair

Step 2: // For User B

$P_{UB} = B_P*P_B$

$U_B = (P_{UB}, P_B)$ //User B key pair

// B_P is the Base Point

Step 3: //Send the Public key of U_B to U_A

Send (P_{UB}, U_B);

Step 4: //Send the Public key of U_A to U_B

Send (P_{UA}, U_A);

Algorithm for textEncryption

Step 1: Calculate $A_{PL} = p*A_P$;

//p = Ascii value of text

//A_P: random point on EC

Step 2: // Calculate kB_P

$kB_P = k*B_P$

//B_P is the Base Point

Step #: // Send Cipher test to receiver, i.e. User B

Cipher Text, $C_M = \{kB_P, A_{PL}+ k * P_{UB}\}$

Algorithm for textDecryption

Let kB_P be the first point

$A_{PL}+ kP_{UB}$ be second point

Step 1: Calculate $P_BkB_P = P_B$ * first_point

//this yields us an equivalent point to kP_{UB}

Step 2: Calculate $A_{PL} = (A_{PL}+ k*P_{UB}) - P_BkB_P$

Now using discrete logarithm concept

Step 3: Evaluate value of sent text from A_{PL}

$A_{PL} = rA_P$

//r is the value to be calculated using the discrete logarithmic concept. r = p, i.e. the original ASCII value.

V. IMPLEMENTATION OF PROPOSED ALGORITHM

Equation of the elliptic curve

$y^2 = x^3 + x + 1 \bmod 131$

Points on the curve can be found as shown in Table I.

TABLE I: SET OF POINTS ON EC

(x, y)	(x, y)	(x, y)	(x, y)	(x, y)	(x, y)	(x, y)
0, 1	21, 47	35, 104	61, 85	88, 112	114, 7	128, 87
0, 130	21, 84	37, 16	62, 44	91, 46	114, 124	
1, 38	22, 45	37, 115	62, 87	91, 85	115, 9	
1, 93	22, 86	39, 4	67, 29	92, 36	115, 122	
2, 50	24, 15	39, 127	67, 102	92, 95	117, 16	
2, 81	24, 116	40, 47	68, 25	96, 37	117, 115	
5, 0	25, 18	40, 84	68, 106	96, 94	120, 10	
7, 58	25, 113	43, 54	70, 47	97, 15	120, 121	
7, 73	26, 7	43, 77	70, 84	97, 116	121, 63	
9, 52	26, 124	48, 48	71, 22	102, 39	121, 68	
9, 79	27, 42	48, 83	71, 109	102, 92	122, 7	
10, 15	27, 89	49, 42	72, 44	105, 52	122, 124	
10,	29, 61	49,	72,	105,	123,	

116		89	87	79	23	
11, 65	29, 70	50, 48	73, 26	107, 63	123, 108	
11, 66	30, 62	50, 83	73, 105	107, 68	124, 31	
12, 13	30, 69	51, 0	75, 0	108, 16	124, 100	
12, 118	33, 48	55, 42	79, 64	108, 115	125, 33	
16, 24	33, 83	55, 89	79, 67	110, 46	125, 98	
16, 107	34, 63	58, 51	86, 20	110, 85	127, 8	
17, 52	34, 68	58, 80	86, 111	113, 35	127, 123	
17, 79	35, 27	61, 46	88, 19	113, 96	128, 44	

Base Point, $B_P = (0, 1)$

Base Point implies, smallest value of co-ordinates satisfying EC over GF(P).

A_P is another affine point on the EC; it may or may not be the same as Base Point, however in order to maintain individual identity, we choose A_P to be different from Base Point.

Let $A_P = (1, 38)$

The ECC method implies that we select an integer k (k<p), which needs to be kept secret. 'p' is the prime value for the EC to be generated upon. Then kG is evaluated by applying a series of additions as discussed before.

Let the sender be known as User A and the receiver be known as User B.

The secret integer k and P_B must be randomly generated, in order to give credibility.

However, for implementation, we shall assume k = 37 and PB =41.

Public key of B

$P_{UB} = P_B * B_P$

$P_{UB} = 41*(0,1) = (43, 77)$

//ENCRYPTION

For a character to be transmitted, User A does the following.

Character = 'T'

ASCII value of character, a = 84

$A_{PL} = 84(1, 38) = (55, 89)$

The coordinates must fit into EC perfectly.

Next we evaluate kP_{UB}, i.e.,

$k*P_{UB} = 37(43, 77) = (125, 33)$

$A_{PL} + kP_{UB} = (55, 89) + (125, 33) = (25, 18)$

$kB_P = 37(0, 1) = (97, 116)$

The encrypted message is

$C_m = \{kB_P, A_{PL}+kP_{UB}\} = \{(97, 116), (25, 18)\}$

kB_P must be included as first coordinate and $A_{PL} + kP_{UB}$ as second coordinate . So that, $kB_P = (x1, y1)$ & $A_{PL} + kP_{UB} = (x2, y2)$.

This concludes the encryption of the text in affine coordinates.

//DECRYPTION

At receiver's end, User B applies his private key P_B on kB_P, i.e. the first co-ordinate. This will in turn provide him with $P_B kB_P = kP_{UB}$. This received co-ordinate is then subtracted from second received co-ordinate,, i.e. $A_{PL} + kP_{UB}$, thus giving the user A_{PL}.

$P_B kB_P = 41 (97, 116) = (125, 33)$

$A_{PL} = \{A_{PL} + kP_{UB}\} - P_B kB_P$

$= (25, 18) - (125, 33) = (55, 89)$

The subtraction is just addition operation with negative y co-ordinate. However the user receives the same slope as is supposed to.

In order to receive the encoded text value (r) from the received A_{PL} coordinate, we use Baby Step Giant Step algorithm to solve the discrete logarithm.

$rA_P = A_{PL}$

where r is to be found.

A_P acts as point P, and A_{PL} acts as point Q

Algorithm BSGS

First we find out the order of the point AP in GF(P)

n = ord (A_P) = 128
then, calculate x = ⌈ √n⌉ = 12
//Calculate baby steps. Ranging from 0 to x-1, i.e. 0- 11.
find R_B = Q - [B]P
where, B = 0 - (x-1) = 0 -11

Step	$R_B = Q - [B]A_P$
0	(55, 89)
1	(62, 87)
2	(128, 87)
3	(37, 115)
4	(21, 84)
5	(27, 89)
6	(110, 85)
7	(97, 15)
8	(114, 7)
9	(55, 42)
10	(86, 111)
11	(34, 68)

// Calculating giant steps, ranging from 1 to x-2
R_B' = [A][x]A_P
where, A = 1 - (x-2) = 1 – 10

Step	R_B' = [A][x]AP
1	(72, 44)
2	(34, 63)
3	(11, 65)
4	(97, 116)
5	(22, 45)
6	(125, 33)
7	(55, 89)
8	(1, 93)
9	(55, 42)
10	(125, 98)

find a and b such that Rb = Rb'
here, as one can notice, A = 7, B= 0
Then,
r = [x] *A + B = 12*7 + 0 = 84
where, r is the required value.
Converting r into its equivalent character encoding gives us
'T', which has the ASCII value 84.

VI. CONCLUSION

In the encryption algorithm proposed here, the parties agree on prime number p, base point B_P, and affine point A_P. As the equation for the elliptic curve used here is general form of Weierstrass equation, hence no need of communicating the values of 'a' and 'b'. However, these values can be replaced, such that, a, b∈ Fp. The security of Elliptic Curve depends on finding the value of k, the larger the number; the harder is it to be solved by hacker. Each communicating party publishes a specific public key for the communication with a specific communicator. With this the receiver is assured that the sender constructed the cipher only because the sender uses receiver's specific public key published for the sender alone and sender's private key for constructing the cipher. This ensures that sender has "digitally signed" the message by using the specific public key published for him alone by the receiver. Hence, the cipher has achieved the qualities confidentiality, and authentication. As each character is encoded into affine point on the EC, it also introduces non-linearity in the character thus hiding its true identity.

Decryption process is itself quite a formidable task, without the knowledge of P_B and A_P; it is impossible to solve the discrete logarithm in order to obtain the encoded message.

REFERENCES

[1] S. Maria Celestin Vigila and K. Muneeswaran, "Implementation of text Based cryptosystem using elliptic curve cryptography", IEEE, 2009.

[2] D. Sravana Kumar, CH. Suneetha and A. Chandrasekhar, "Encryption of data using Elliptic Curve over Finite Field", IJDPS, Vol. 3, No. 1, 2012.

[3] R. Rajaram Ramasamy, M. Amutha Prabakar, M. Indra Devi and M.Suguna, "Knapsack based ECC encryption and decryption", International Journal of Network Security, Vol. 9, No. 3, PP. 218-226,Nov. 2009.

[4] Padma Bh, D. Chandravathi and P. Prapoorna Roja, "Encoding and Decoding of a Message int the Implementation of Elliptic Curve Cryptography using Koblitz's method", IJCSE, Vol. 02, No. 05, 2010.

[5] William Stallings, Cryptography and Network Security, Prentice Hall, 5th Edition, 2010.

[6] Vivek Kapoor, Vivek Sonny Abraham and Ramesh Singh, "Elliptic Curve Cryptography", ACM Ubiquity, vol. 0, Issue 20, May 20-26, 2008.

[7] Darren Hankerson, Julio Lopez Hernandez and Alfred Menezes, "Software implementation of Elliptic Curve Cryptography over Binary Fields", Cryptographic Hardware and Embedded Systems — CHES 2000, Lecture Notes in Computer Science Volume 1965, 2000, pp 1-24.

[8] D.J Malan, M. Welsh and M.D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on* , vol., no., pp.71,80, 4-7 Oct. 2004

[9] M.Aydos, T.Yanik and C.K.Kog, "High-speed implementation of an ECC based wireless authentication protocol on an ARM microprocessor," *lEE Proc Commun.,Vol.* 148, No.5, pp. 273-279, October 2001.

[10] Kristin Lauter, "The Advantages of Elliptic Cryptography for Wireless Security", *IEEE Wireless Communications*, pp. 6 2 - 67, Feb. 2006.

