Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 9, September 2016, pg.81 – 88

Prime Field over Elliptic Curve Cryptography for Secured Message Transaction

Md. Yosuf Zamil[1], Ditee Yasmeen[2]

[1]B.Sc (Hon's.), Department of Computer Science and Engineering, Institute of Science and Technology, National University, Bangladesh
[2]Assistant Professor, Department of Computer Science and Engineering, Institute of Science and Technology, National University, Bangladesh
Email: [1] yosuf.zamil@gmail.com, [2] ditee.yasmeen@yahoo.com

----------ABSTRACT----------

An improved model has been proposed in this research paper for both session-key distribution and electronic message transactions through Prime Field Elliptic Curve Cryptography (PFECC). Frequent key changes are very much desirable for secure electronic communications in symmetric key encryption. Because it is needed to limit the amount of data compromised if an intruder or attacker learns the communicating keys. In public key cryptography, there is an important factor, which is the key length. It is optimal to keep smaller key size but gain large security. For imposing the better security, each time of communication a temporary key, called session-key, is used for symmetric key cryptography. Session-key distribution is the process of delivery a key to two parties who wish to exchange data without allowing others to know the key. For two parties A and B, a key distribution technique using Key Distribution Center (KDC) has been chosen. Here a trusted third party (KDC) has introduced to secretly distribute the base point of the Elliptic Curve, among the communicating parties. For session-key distribution, KDC acts as the sender and the rest of the communicating parties under the domain of the KDC acts as the receiver. On the contrary, for electronic message transactions, the KDC only involves in the base point distribution. Rests of the communications are done between two individual parties, under the domain of that KDC. This model may be applied to encrypt all kind of electronic messages, such as e-mail, sms, session key and etc.

Keywords: ECC, Public key cryptography, Prime Field, KDC, Session key.

I. INTRODUCTION

ECC is a public key cryptography which has public and private keys for authentication. The utilization of elliptic curves in cryptography has been proposed for the first time by Koblitz and Victor Miller individually in mid 1980s [1]. ECC is known as a sort of PKC which is built upon algebraic structure of elliptic curve over finite fields [1]. Difficulty of elliptic curve discrete logarithm problem (ECDLP) plays a major role in the security of ECC, and this problem can be resolved in exponential time [2]. Meanwhile it has to be added that performance of this algorithm is mainly intertwined with the efficiency of its scalar multiplication algorithm [5]. Hamming weight of the private key is a determinant factor in algorithm efficacy regarding scalar arithmetic level of the computation [3]. Hamming weight is defined as a means to measure the number of none zero digits in a scalar representation. As the extent of Hamming weight lowers, the speed of scalar multiplication performance rises up. Accordingly, scalar recoding method can be used

to lessen Hamming weight of scalar representation of private key. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user knows the private key whereas the public key is distributed to all users taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. '*Domain parameters'* in ECC is an example of such constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography.

Understanding ECC needs full mathematical background on elliptic curves. The general cubic equation of elliptic curves is $y^2+axy+by=x^3+cx^2+dx+e$. But for our purpose it is sufficient to limit the equation to the form $y^2 = x^3 + ax + b$ where $4a^3 + 27b^2 \neq 0$. Let Ep(a,b) be the set consisting of all the points (x, y) that satisfy the above equation together with element at infinity O. An abelian group can be defined based on the set Ep(a, b) over addition operation for specific values of a and b[4]. If P, Q and R are points on Ep(a, b) the relations commutativity, associativity, existence of an identity element and existence of inverse hold good. Such a group can then be used to create an analogue of the discrete logarithm problem which is the basis for ElGamal public key cryptosystems. Each value of the 'a' and 'b' gives a different elliptic curve. The public key is a point in the curve and the private key is a random number in the interval [1, n-1], 'n' is the curve's order. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G is the point on the curve. The generator point G, the curve parameters 'a' and 'b', together with few more constants constitutes the domain parameters of ECC. [5]

II. ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. But it is relatively easy to find Q where k and P are known. k is the discrete logarithm of Q to the base P. Thus, point multiplication is the basic operation in ECC. For example, the multiplication of a scalar 'k' with any point 'P' on the curve in order to obtain another point 'Q' on the curve.[3]

III. PROPOSED SYSTEM OF ECC

A. Role of Key Distribution Center

A key distribution center is developed, who will responsible for distributing **secret base point** to pairs of users as needed. Each user must share a unique key with the key distribution center for purposes of base point distribution.

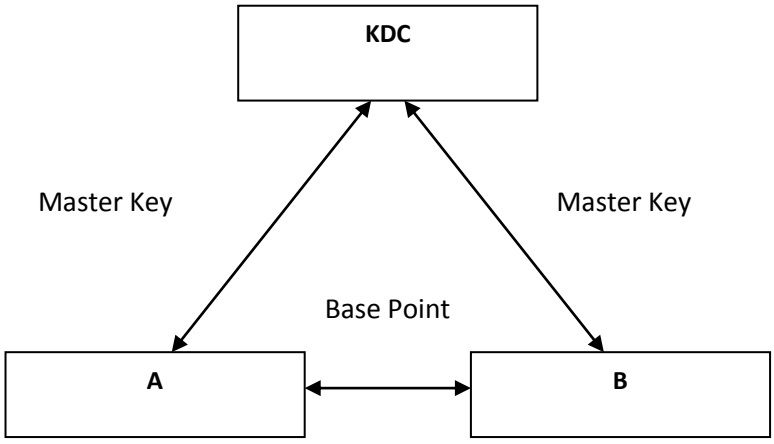

Figure 1: Base Point Distribution

At least two levels of transactions must be used:

- Communication between end systems for establishing an ECC protocol used a temporary base point, chosen from the elliptic curve.

- Base Points are transmitted in encrypted form, using a master key that is shared by the KDC and an end system or user.

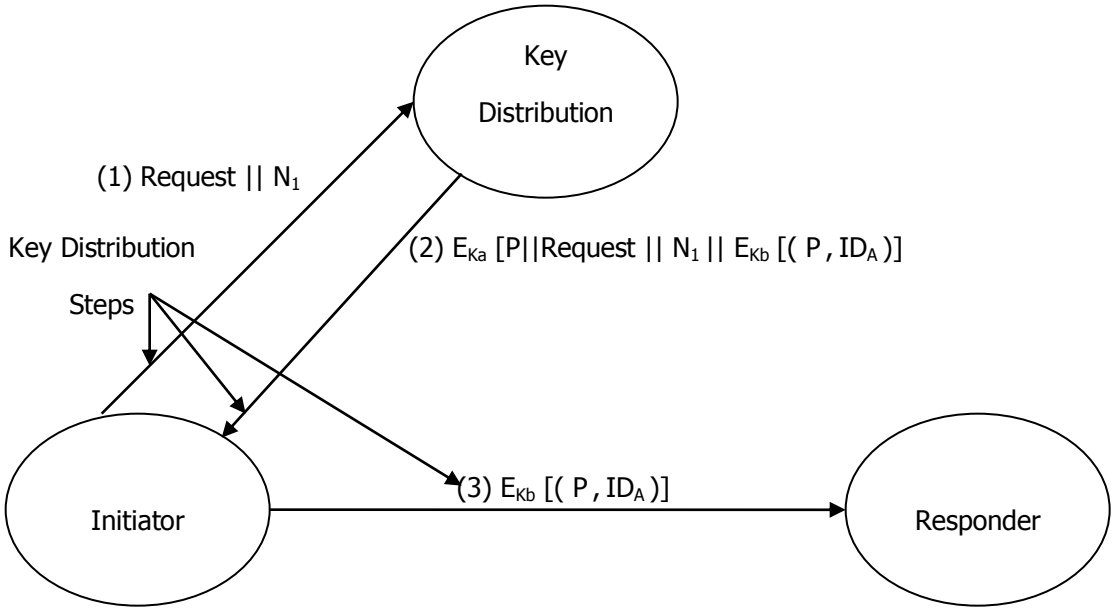

Figure 2: Base Point Distribution Scenario using KDC

Step 1: Session Key Request

- A issues a request to KDC: Request (B) || N_1.
  - N_1 is a number used once only (Nonce), usually chosen as a random number of sufficiently large.
  - The ID of B is part of the request.

Step 2: Issue of Base Point

- KDC sends base point and other information back to A, encrypted by A's master key.
  - P: Base Point.
  - Request (B): Tell A this what it asked for.
  - N_1: Nonce sent by A initially.
  - EK_b (P,IDA): To be forwarded by A to B.

Step 3: Forward Base Point

- A sends the information from KDC to B.
  - EK_b(P,IDA).

Each user must share a unique master key with the KDC. If there are N end users, N(N -1)/2 base points are needed at any one time, but only N master keys are required. The master key can be distributed in a non-cryptographic way, such as physical delivery.

Now, for large networks, a single KDC may not be adequate. A hierarchy of KDCs can be established where each local KDC is responsible for a small domain of the overall network. If the two parties of an exchange are within the same local domain, their local KDC is responsible for base point distribution. Otherwise, the corresponding local KDCs can communicate through a global KDC. Any one of the three KDCs involved can select the base point.

B. System Model

An underlying finite field Fp is chosen. An elliptic curve E defined over Fp, and a base point P on E are chosen. The order of the point P is denoted by n. The field Fp and curve E, comprise the system parameters, and are public information. The point P is chosen by the Key Distribution Centre (KDC) and distributes secretly to the sender and receiver for the current communication.

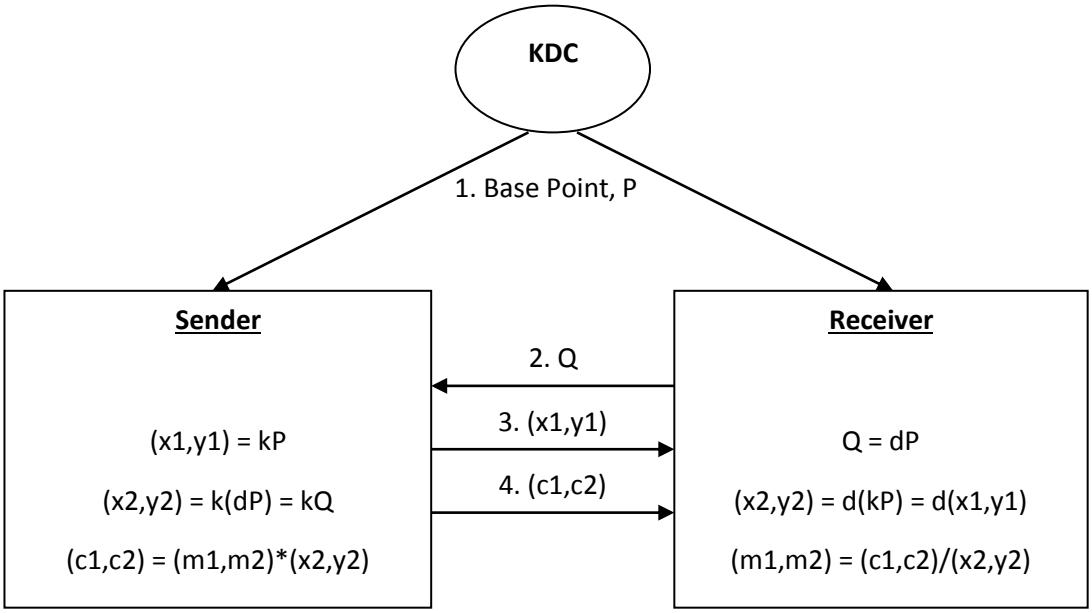

Figure 3: Proposed System Model

C.Key Generation
Entity B (receiver) shall perform the following operations.

1. Select a random integer d in the range [1, n-1].
2. Compute the point Q = dP.
3. The entity's public key consists of the point Q.
4. The entity's private key is the integer d.

D.Encryption Process
(Entity A sends a message M to entity B). Entity A performs the following steps:

1. Look up B's public key: Q.
2. Represent the message M as a pair of field elements (m1, m2), m1ε Fq, m2 ε Fq .
3. Select a random integer k in the range [1, n-1].
4. Compute the point (x1, y1) = kP.
5. Compute the point (x2, y2) = kQ.
6. Combine the field elements m1, m2 and x2, y2 in a predetermined manner to obtain two field elements c1 and c2.
7. Transmit the data c = (x1, y1, c1, c2) to B.

E.Decryption Process
(Entity B decrypts cipher text c = (x1, y1, c1, c2) received from A). Entity B performs the following steps:

1. Compute the point (x2, y2) = d (x1, y1), using its private key d.
2. Recover the message m1 and m2 from c1, c2 and x2, y2.

IV.EXAMPLE of PFECC

We know the equation, $y^2 + axy + by = x^3 + cx^2 + dx + c$
But we have to calculate on partial equation, $y^2 \bmod p = (x^3 + ax + b) \bmod p$
Where a and b are integers number p is a prime number
Let, p = 5
We need to calculate $(4a^3 + 27b^2) \bmod p != 0$, if a = 2 and b = 4

(32+27*16) mod 5 != 0
464 mod 5 != 0
2 != 0

Table 1: Points calculation table for PFECC

X	$y^2 \bmod p$	=	$(x^3 + ax + b) \bmod p$	Points
When x=0	$y^2 \bmod 5$	=	4 mod 5 = ±2	(0, 2), (0, 3)
x=1	$y^2 \bmod 5$	=	7 mod 5 = ±√2	No Solution
x=2	$y^2 \bmod 5$	=	16 mod 5 = ±1	(2, 1), (2, 4)
x=3	$y^2 \bmod 5$	=	37 mod 5 = ±√2	No Solution
x=4	$y^2 \bmod 5$	=	76 mod 5 = ±1	(4, 1), (4, 4)

All points: (0, 2), (0, 3), (2, 1), (2, 4), (4, 1), (4, 4), (∞, ∞)

Table 2: All points with number of character message

N	Points	Letters
1	(0, 2)	A
2	(0, 3)	B
3	(2, 1)	C
4	(2, 4)	D
5	(4, 1)	E
6	(4, 4)	F
7	(∞, ∞)	Points of ∞

Here, Co-prime, n = 1 to p – 1, p = 5
We got the number of co-prime, 1, 2, 3
Let, the Base Point P (0, 2). The public key distributed by **KDC** (Key Distribution Centre).

User A:

Private Key k = 3

a) Compute the first points, (x1,y1) = kp = 3(0, 2) = (2, 1)
(x1, y1) send to the User B.

b) Compute the second points, (x2,y2) = kQ = k(dp)= 3(2(0, 2))
= 6(0, 2)
= (4, 4)

c) Calculate the Cipher text, (c1, c2) mod p = [(m1, m2) * (x2, y2)] mod p
⇒ (c1, c2) mod 5 = [(m1, m2) * (x2, y2)] mod 5
Suppose we want to send a character (message) from the table (m1, m2) = (2, 4) = D
= [(2, 4) * (4, 4)] mod 5
= (8, 16) mod 5
∴ (c1, c2) mod 5 = (3, 1)

We can send (x1, y1) and (c1, c2) to the User B.

User B:

Private Key d = 2

a) Compute the first points, Q = dp
= 2(0, 2)
= (0, 3)

Q send to the User A.

b) Compute the second points, (x2, y2) = d(x1, y1)
= d(kp)
= 2(3(0, 2))
= 6(0, 2)
= (4, 4)

c) Gaining the plaintext, (m1, m2) mod p = $\frac{(c1,c2)}{(x2,y2)}$ mod p

⇒ (m1, m2) mod 5 = $\frac{(3,1)}{(4,4)}$ mod 5
⇒ (4m1, 4m2) mod 5 = (3, 1) mod 5

We can compute (m1, m2) using Congruent Module, a ≡ b mod m
= a-b/m, where fully divisible

4m1 mod 5 = 3 mod 5
= $\frac{4m1-3}{5}$
Let, m1 = 2
= $\frac{4*2-3}{5}$
= $\frac{5}{5}$
= 1 (Fully Divisible)

4m2 mod 5 = 1 mod 5
= $\frac{4m2-1}{5}$
Let, m1 = 4
= $\frac{4*4-1}{5}$
= $\frac{15}{5}$
= 3 (Fully Divisible)

∴ (m1, m2) = (2, 4) = D

Finally we got the original character message 'D' while we decrypt the cipher text.

V. IMPLEMENTATION OF THE PROPOSED SYSTEM

Giving the Prime Number and Checked the Equation

Enter your prime number: 43
43 Thanks this is a Prime Number
Enter the value of 'a' and 'b' for Check (4a3+27b2) mod P ≠ 0:
2 4
(4a3+27b2) mod P ≠ 0 Equation is Satisfied.
When 0 then y2 = 4.0 and points is (0, 2) and (0, 41)
When 2 then y2 = 16.0 and points is (2, 4) and (2, 39)
When 8 then y2 = 16.0 and points is (8, 4) and (8, 39)

When 12 then y2 = 36.0 and points is (12, 6) and (12, 37)
When 16 then y2 = 4.0 and points is (16, 2) and (16, 41)
When 27 then y2 = 4.0 and points is (27, 2) and (27, 41)
When 33 then y2 = 16.0 and points is (33, 4) and (33, 39)
When 42 then y2 = 1.0 and points is (42, 1) and (42, 42)

Points (x, y) for every characters

All points for X and Y in below:

No	Points(X,Y)	Character
1	(0, 2)	a
2	(0, 41)	b
3	(2, 4)	c
4	(2, 39)	d
5	(8, 4)	e
6	(8, 39)	f
7	(12, 6)	g
8	(12 , 37)	h
9	(16, 2)	i
10	(16, 41)	j
11	(27, 2)	k
12	(27, 41)	l
13	(33, 4)	m
14	(33, 39)	n
15	(42, 1)	o
16	(42, 42)	p

Here, Co-prime, n = 1 to p – 1, p = 43

We got the number of co-prime: 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, and 43

User 'A' private key and calculate the corresponding point

Enter User A private key (K): 5
Base point P (0, 2) and then USER A

a) Compute (x1, y1) = KP = (8, 4)
b) User A compute (x2, y2) = KQ = K (DP) = (42,1)

User 'B' private key and calculate the corresponding point

Enter User B private key (D): 3
Base point P (0, 2) and then USER B

a) Compute Q = DP = (2, 4)
a) b) User B compute (x2, y2) = D(x1, y1) = D (KP) = (42, 1)

Choose the character message what we want to encrypt

Enter a Character message from the table to Encrypt: d
Corresponding two points of 'd' is: 2 and 39

Now we calculate Cipher Text (c1, c2) mod 43 = [(m1, m2) * (x2, y2)] mod P

=> (c1, c2) mod 43 = (2, 39) * (42, 1) mod 43
=> (c1, c2) mod 43 = (84, 39) mod 43
So, (c1, c2) mod 43 = (41, 39)

Now (x1, y1) and (c1, c2) send to the User B.

Gaining the character message while we decrypt

Gaining Plain text using (m1, m2) mod 43 = (c1, c2)/(x2, y2) mod 43

=> (m1, m2) mod 43 = (41, 39) / (42, 1) mod 43

=> (42 m1, 1m2) mod 43 = (41, 39) mod 43

=> 42m1 = 41 mod 43 AND 1m2 = 39 mod 43

We know, Congruent module => a -= b mod m

(a-b) /m that is fully divisible.

So, (42m1 - 41)/43 AND (1m2 - 39)/43

Enter m1 and m2 value where this equation fully divisible:

2 39

Decrypted Character Message is ‘d’.

VI. CONCLUSION

The first result is that ECC algorithm is faster than RSA algorithm for encrypt and decrypt messages and it uses smaller key size length. One of the most important problems in smart cards is limitation of storage. Providing a multipurpose smart card with smaller key size is very important. On the other hand, since in the multipurpose smart card there are three different systems, high speed is one the most important requirements in the system. ECC algorithm has high level of security and it is faster than RSA algorithm. Since, in the passport card biometric verification is used for authentication of the person and it is a contact-less card, ECC algorithm is the best algorithm for passport system.

REFERENCES

[1] Sonali U. Nimbhorkar, and Dr. L. G. Malik. “A Survey On Elliptic Curve Cryptography (ECC)” International Journal of Advanced Studies in Computers, Science and Engineering vol.1, 2012, issue 1 pp. 1-5.

[2] Koblitz, N. Elliptic curve cryptosystems. Mathematics of computation. No. 48, pp. 203-209,1987

[3] https://eprint.iacr.org/2008/099.pdf

[4] Menezes, J., Van Oorschot, P. C., and Vanstone, S. A., "Handbook of Applied Cryptology" CRC Press, LLC 1997.

[5] Hankerson, D., L`opez Hernandez, J. and Menezes, A., Software implementation of elliptic curve cryptography over binary fields. Cryptographic Hardware and Embedded Systems, CHES’00. LNCS,1965:1–24, 2000.

