Diffie-Hellman Key Agreement with Elliptic Curve Discrete Logarithm Problem

Samta Gajbhiye
SSGI, SSTC,
Junwani, Bhilai ,CG, India

Sanjeev Karmakar
BIT, Bhilai
CG, India

Monisha Sharma
SSGI, SSTC, Junwani
Bhilai, CG, India

ABSTRACT

Since the invention of public -key cryptography, numerous public -key cryptographic systems have been proposed. Each of these systems relies on a difficult mathematical problem for its security. Today, three types of systems, classified according to the mathematical problem on which they are based, are generally considered both secure and efficient. The systems are:the integer factorization systems (of which RSA is the best known example), the discrete logarithm systems (such as the U.S. Government's DSA), the elliptic curve discrete logarithm systems (also known as *elliptic curve* cryptosystems).

This paper focuses on implementing cryptographic services with elliptic curve cryptography (ECC). The principle attraction of ECC is that it appears to offer equal security for a far smaller key size, thereby reducing processor overhead. This paper implements Diffie –Hellman Key aggrement Procotocol using Elliptic Curve as the mathematical technique over prime field F_p

Keywords

Diffie-Hellman key Agreement protocol. Elliptic curve cryptography, Elliptic Curve Diffiee Hellman(ECDH)

1. INTRODUCTION

Elliptic Curve Cryptography (ECC) was independently proposed by koblitz [1] and Miller [2] in the late 1980s. ECC is a public key cryptographic scheme that uses the properties of Elliptic Curves in mathematics to develop cryptographic algorithms. Security of ECC is based on the intractability of Elliptic Curve Discrete Logarithm Problem (ECDLP) [3]. ECC is defined by Elliptic Curve domain parameters given by: *T = (q,FR,a,b,c,G,n,h)* where; q: the prime p or 2m that defines the curve's form, FR: the field representation, a, b: curve coefficients, G: the base point (Gx, Gy), n: the order of G, which must be a large prime, and h: the cofactor co-efficient. The basic advantage of using elliptic curves for cryptography purpose is that it appears to provide equal security for a far smaller key size, and thus reducing processing overhead[4].

Diffie-Helmann (DH) key exchange protocol is the first public key cryptography scheme, and it was proposed by Witfield Diffie and Martin Hellman in 1976. This protocol uses a pair of keys (secret and private keys), since it is a public key cryptographic scheme[5]. DH key exchange protocol is based on the difficulty of computing logarithmic functions of prime exponents, and this is known as Discrete Logarithm Problem (DLP). But Diffie-Hellman problem over elliptic curve with small keys is much harder to solve than the discrete logarithm over finite fields[5]

Rest of the paper is organized as follows. Section 2 and 3 discusses the background of Elliptic Curve Cryptography and Elliptic Curve Diffie Hellman Key Exchange Protocol. Section 4 shows the implementation results ECDLP-based Diffie Hellman. Section 4 is Results and Discussions and finally, Section 5 is conclusion with application and future scope.

2. MATHEMATICS BEHIND ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curves are mathematical constructs that have been studied by mathematicians since the seventeenth century. In 1985, Neal Koblitz and Victor Miller independently proposed public –key systems using a group of points on an elliptic curve, and elliptic curve cryptography (ECC) was born[1,2]. ECC delivers the highest strength per bit of any known public -key system because of the difficulty of the hard mathematical problem ie Elliptic Curve Discrete Logarithm problem (ECDLP) upon which it is based. Given the best known algorithms to factor integers and compute elliptic curve logarithms, the key sizes are considered to be equivalent strength based on MIPS years needed to recover one key[6,7]

An elliptic curve E over a field K is defined by an equation (Weierstrass equation)[9, 10]

$E: y^2 = x^3 + ax + b$ …… (1)

where x, y are elements of GF(p),

$a, b \in K$ and $\Delta \neq 0$,

'p' is known as modular prime integer making the EC finite field

Δ is the determinant of E and is defined as follows:

determinant$-16(4a^3 + 27b^2) \neq 0 (\text{mod } p)$. ….(2)

2.1 Point Addition

The addition rule is best explained geometrically. Let P = (x_1, y_1) and Q = (x2, y2) be two distinct points on an elliptic curve E. Then the sum R, of P and Q, is defined as follows. First draw a line through *P* and *Q*; this line intersects the elliptic curve at a third point. Then *R* is the reflection of this point about the *x*-axis. This is depicted in Figure 1.

Mathematically addition is defined as:

Let $P = (x_1, y_1) \in E(K)$ and $Q = (x_2, y_2) \in E(K)$, where $P \neq \pm Q$.

Then $P + Q = (x_3, y_3)$,

Where $x_3 = \{(y_2 - y_1) /(x_2 - x_1)\}^2 - x_1 - x_2$ ...(3)

$y_3 = \{(y_2 - y_1)/ (x_2 - x_1)\}(x_1 - x3) - y_1$. ………(4)

2.2 Point Doubling

Similarly the double *R*, of *P*, is defined as follows. First draw the tangent line to the elliptic curve at *P*. This line intersects the elliptic curve at a second point. Then *R* is the reflection of this point about the *x*-axis. This is depicted in Figure 2.

Mathematically doubling is defined as:

Let $P = (x1, y1) \in E(K)$, where $P \neq -P$.

Then $2P = (x_3, y_3)$,

Where $x_3 = \{(3x1^2+a\ y_1)/2y_1\}^2 - 2x_1$ ……….(5)

$y_3 = \{(3x_1^2+a\ y_1)/2y_1\}^2 (x_1 - x_3) - y_1$. ………(6)

When the elliptic curve group is described using additive notation, the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number k such that Pk = Q . And k is large enough such that it would be infeasible to determine k. The value of kP can be calculated by a series of doubling and addition operation

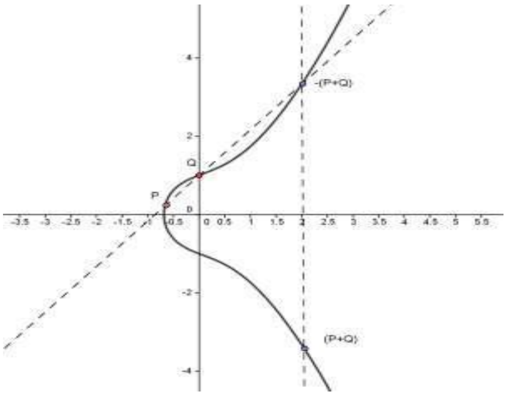


Fig1:Point Addition

When the elliptic curve group is described using additive notation, the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number k such that Pk = Q . And k is large enough such that it would be infeasible to determine k. The value of kP can be calculated by a series of doubling and addition operation.

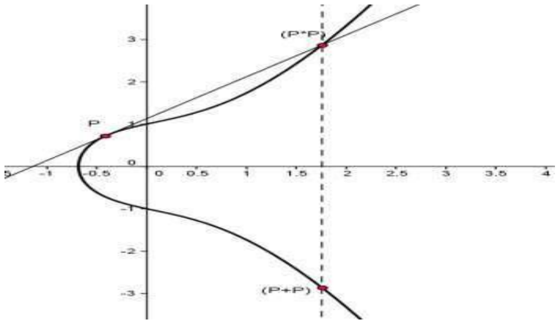


Fig 2: Point Doubling

3. ECDLP— DIFFIEE- HELLMAN KEY AGREEMENT PROTOCOL

In order to generate a shared key between Alice and Bob using ECDH key exchange protocol, both Alice and Bob should agree beforehand to use the same Elliptic Curve domain parameters [8,9] .The following procedure allows Alice and Bob to securely exchange the value of a point on an elliptic curve, although neither of them initially knows the value of the point:

- Alice (sender) computes key $k = (x_K, y_K) = d_{Alice} * Q_{Bob}$, where d_{Alice} is Alice's private key an Q_{Bob} is Bob's public key
- Bob (receiver) computes key $l = (x_L, y_L) = d_{Bob} * Q_{Alice}$,where d_{Bob} is Bob's private key, and Q_{Alice} is Alice's public key
- Since $d_{Alice} * Q_{Bob} = d_{Alice}\ d_{Bob}\ G = d_{Bob}\ d_{Alice}\ G = d_{Bob} * Q_{Alice}$, then k = l, hence $x_K = x_L$.
- Hence the shared key is x_K .

4. RESULTS AND DISCUSSION

Snap shots shows the results with p=29, a=4 and b=20

Then the elliptic curve expression is E: $y^2 = x^3 + 4x + 20$ defined over F_{29}. Also since $\Delta = -16(4a3 + 27b2) = -176896 \neq 0$ *(*mod 29*)*, so *E* is indeed an elliptic curve. Thirty Seven coordinates that satisfy *E(*F_{29}*)* are: ∞ (0,7) (0,22) (1,5) (1,24) (2,6) (2,23) (3,1) (3,28) (4,10) (4,19) (5,7) (5,22) (6,12) (6,17) (8,10)(8,19) (10,4) (10,25) (13,6) (14,6) (14,23) (15,2) (15,27) (16,2) (16,27) (17,10) (17,19) (19,13) (19,16) (20,3) (20,26) (24,7) (24,22) (27,2) (13,23) (27,27)

Generator or base points are: (0,7) (0,22) (1,5) (1,24) (2,6) (2,23) (3,28) (4,10) (4,19) (5,7) (5,22) (6,12) (6,17) (8,10) (8,19) (10,4) (10, 25) (13, 6) (13,23) (14,6), (14,23) (15,27) (16,2) (16,27) (17,10), (17,19) (19,16) (20,3) (20,26) (24,7) (24,22) (27,27).

The coordinates and base points are shown in the figure 3

Following steps illustrates computation of shared key:

Step-1: Let Alice and Bob agrees on same base point (3, 28) , calculates their public key and

send it to each other

Step-2: Let Alice selects its private key d_{Alice} i.e n as 20. Then Alice Public Key i.e

Q_{Alice} (x_A, y_A) = 20 * (3,28) = (15,2)

Step-3: Let Bob selects its private key d_{Bob} i.e n as 15. Then Bobs Public Key i.e

Q_{Bob} (x_B, y_B) = 15 * (3,28) = (20, 26)

Step-4: Alice sends Q_{Alice} ie (x_A, y_A) to Bob and Bob sends $Q_{Bob\ ie}$ (x_B, y_B) to Alice.

Step-5: Alice computes shared key as

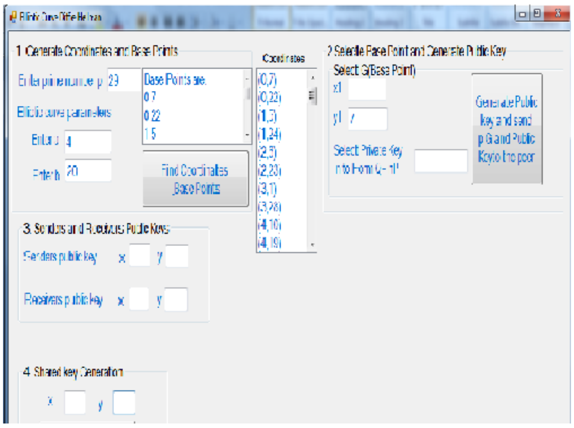
$P_{AB} = d_{Alice} * Q_{Bob} = d_{Alice}\ d_{Bob}\ G$ = 20 * (20,26)

Step-6: Bob computes shared key as

$P_{BA=}\ d_{Bob} * Q_{Alice} = d_{Bob}\ d_{Alice}\ G$ = 15 * (15,2)

Step-7: $P_{AB\ =}\ P_{BA=\ (5,22)}$

Step 4 to Step 6 is shown in Figure 4



Elliptic Curve Diffie Hellman

1 Generate Coordinates and Base Points

Enter prime number p 29

Elliptic curve parameters

Enter a 4

Enter b 20

Base Points are:
0 7
0 22
1 5

Coordinates
(0,7)
(0,22)
(1,5)
(1,24)
(2,6)
(2,23)
(3,1)
(3,28)
(4,10)
(4,19)

2 Select Base Point and Generate Public Key

Select G(Base Point)

x1

y1 7

Select Private Key n to Form Q = nP

3 Senders and Receivers Public Keys

Senders public key x y

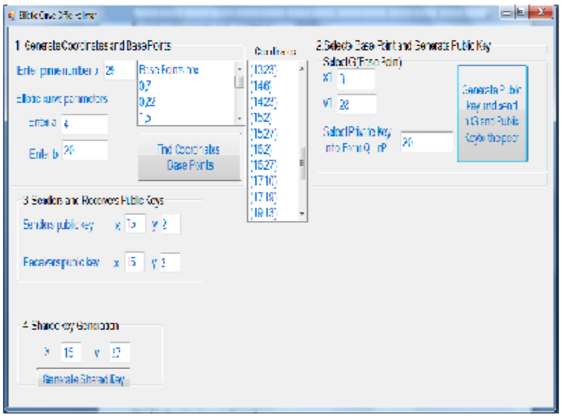
Receivers public key x y

4 Shared key Generation

x y

Find Coordinates Base Points
Generate Public key and send p G and Public Key to the peer

Figure 3: Cordinates and base point of E: E: $y^2 = x^3 +4x +20$ over F_{29}



Elliptic Curve Diffie Hellman

1 Generate Coordinates and Base Points

Enter prime number p 29

Elliptic curve parameters

Enter a 4

Enter b 20

Base Points are:
0,7
0,22
1,5

Coordinates
(13,23)
(14,6)
(14,23)
(15,2)
(15,27)
(16,2)
(16,27)
(17,10)
(17,19)
(18,13)

2 Select Base Point and Generate Public Key

Select G(Base Point)

x1 1

y1 28

Select Private key n to Form Q = nP 20

3 Senders and Receivers Public Keys

Senders public key x 15 y 2

Receivers public key x 15 y 2

4 Shared key Generation

x 15 y 27

Find Coordinates Base Points
Generate Public key and send n G and Public Key to the peer
Generate Shared Key

Figure 4: Shared key of Alice and Bob $P_{AB} = P_{BA}$

Data encryption and secure communication can occur, once secure exchange of the symmetric key is complete

5. CONCLUSION AND FUTURE SCOPE

The Diffie–Hellman scheme is one of the exchanging key cryptosystem, no messages are involved in this scheme nor in this report, and we try to benefit from this scheme by using the key (which exchange it) as a secret key. The longer a symmetric key is in use, the easier it is to perform a successful cryptanalytic attack against it. Therefore, changing keys frequently is important. Both sides of the communication still have the shared secret and it can be used to encrypt future keys at any time and any frequency desired. The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack (MITM). And attacker can decrypts, read and modify any messages sent out by Alice or Bob, This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants.

ECDH can be applied on devices where limited processing power and limited memory capacity exist.

6. REFERENCES

[1] N. Koblitz. “Elliptic Curve Cryptosystems”. Mathematics of Computation, Vol. 48, pp 203-209, 1987

[2] V. Miller. “Uses of Elliptic Curves in cryptography”. CRYPT’85, LNCS 218, pp 417-426, 1986.

[3] P. Bulens, G. M. de Dormale and J. J. Quisquater. “Hardware for collision search on Elliptic Curve over GF (2m)”. SHARCS, Ecrypt Workshop, 2006.

[4] Williams Stallings, Cryptography and Network Security, Prentice Hall, 4th Edition, 2006

[5] Diffie and M. E. Hellman. “New directions in cryptography”. IEEE Transactions on Information Theory, Vol. 22, No. 6, pp 644-654, 1976.

[6] Harper, G., Menezes, A., and Vanstone, S., “Public -key cryptosystems with very small key lengths,” Advance in Cryptology, EUROCRYPT ’92 - Lecture Notes in Computer Science, Volume 658, Springer-Verlag, pages 163-173, 1993.

[7] D. Hankerson, A. Menezes and S. Vanstone. Guide to Elliptic Curve Cryptography. Springer-Verlag, New York, NY, 2004. (book)

[8] M. S. Anoop. “Elliptic curve cryptography – an implementation tutorial”. Technical Report, Tata Elxsi Ltd, 2007.

[9] S. Kumar, et al. “Embedded end-to-end wireless security with ECDH key exchange”. In Proceedings of 46th IEEE International Midwest Symposium on Circuits and Systems, 2003

[10] Koblitz, N., A Course in Number Theory and Cryptography, Springer-Verlag, Second Edition,1994.(book)