

A Diffie-Hellman key exchange for self-Encryption over points on the Elliptic Curve Cryptography

B.Ravi Kumar¹, A.Chandra Sekhar², G.Appala Naidu³

¹Department of Mathematics, Gitam University, Visakhapatnam, India;

²Department of Mathematics, Gitam University, Visakhapatnam, India;

³Department of Mathematics, Andhra University, Visakhapatnam, India.

(Received July 23, 2016, accepted January 15, 2017)

Abstract. Cryptography is the combination of Mathematics and Computer Science. Cryptography is used for encryption and decryption of data using mathematics. Cryptography transmit the information in an illegible manner such that only intended recipients will be able to decrypt the information. In the recent years, researchers developed several new encryption methods. Among such Diffie–Hellman encryption is the one laid a concealed platform for the researchers in Cryptography. Ever since several mathematical models were applied for encryption/decryption. In this paper, we introduced a Diffie–Hellman encryption, which uses points on the elliptic curve, and as an additional security the Fibonacci Q-matrix is introduced.

Keywords: Diffie-Hellman; Fibonacci sequence; encryption; decryption.

1. Introduction

In the year, 1985 Victor Miller and Neal Koblitz first introduced the Elliptic curve cryptography. Elliptic curve cryptography has proven its security by with standing for a generation of attacks. In the recent years, as the wireless communication has grown rapidly, the numerous companies have adopted Elliptic curve cryptography as an innovative security technology. Elliptic curve employs a relatively short encryption key and the shorter key size is faster and requires less compelling power than the other requires. Elliptic curve cryptography, encryption key provides the same security as ‘1024’-bit RSA encryption key [1][2][3][4][8].

In general, cubic equations for elliptic curves take the following form, known as Weierstrass equation [6]: $y^2 + gxy + hy = x^3 + ix^2 + jx + k$.

Where g, h, i, j, k are real numbers and x, y take on values in the real numbers. For our purpose, it is sufficient to limit ourselves to equations of the form: $y^2 = x^3 + gx + h$.

Where x, y, g, h belong to R, Q, C or F_p . Also include in the definition of an elliptic curve is a single element denoted by O and called the point at infinity or the zero point. There is also a requirement that the discriminant $\Delta = 4g^3 + 27h^2$ [4][5][7].

2. Diffie-Hellman algorithm

Alice wants to send a message to Bob using elliptic curve Diffie-Hellman encryption/decryption scheme. Choose the point $C(x_1, y_1)$ on the elliptic curve. Alice selects a private key ‘ K_A ’ and generates the public key $P_A = K_A C$ and Bob selects a private key ‘ K_B ’ and generates the public key $P_B = K_B C$ [9,10,11].

2.1. Encryption

Alice should do the following:

Step 1: Alice selects the Bob public key $P_B = K_B C$.

Step 2: Compute $K_A P_B = K_A (K_B C)$.

Step 3: Represent the message as an integer M in the interval $[0, p-2]$, m is a point on E .

Step 4: Compute $CT_m = M + K_A (K_B C)$.

Step 5: Send the encrypted message ‘ CT_m ’ Bob publicly.

2.2. Decryption

To recover the plaintext ‘ M ’ from C Bob should do following:

Step 1: First Bob selects the Alice public key $P_A = K_A C$.

Step 2: Compute $K_B (K_A C)$.

Step 3: Now Bob computes the inverse element of $K_B (K_A C)$ is $-K_B (K_A C)$.

Step 4: Recover $M = M + K_A K_B C - K_A K_B C$.

3. Proposed algorithm

Alice wants to send a message to Bob using elliptic curve Diffie-Hellman encryption. Alice chooses the elliptic curve $y^2 = x^3 + gx + h$ over the field Z_p .

Choose the point G on the elliptic curve. Alice selects a private key 'a' and generates the public key $A = aG$ and Bob selects a private key 'b' and generates the public key $B = bG$.

3.1. Encryption

Step 1: Alice selects the Bob's public key $B = bG$.

Step 2: Alice uses her own private key 'a' and Computes $B = a(bG)$.

Step 3: Alice chooses a random integer k , where $1 \leq k \leq p-2$.

Step 4: Compute $k(abG)$.

Step 5: Alice wants to send the message q_i to Bob.

Step 6: Alice wants to convert the message into the points on the elliptic curve. She chooses a point Q , which is a generator of the elliptic curve. By using ASCII characters of upper case letter into the points on the elliptic curve.

Let $A = \{1P, 2P, 3P, \dots, 255P\}$ and $B = \{\text{ASCII characters}\}$.

Alice defines one to one correspondence $f: A \rightarrow B$ by $f(np) = x_n$, where $n = 1, 2, \dots, 255$ and $\{x_1, x_2, x_3, \dots, x_{255}\}$ are the ASCII characters.

Step 7: To create 2×2 matrix with entries, are the points on the elliptic curve.

$m_1 = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$, $m_2 = \begin{pmatrix} a_5 & a_6 \\ a_7 & a_8 \end{pmatrix}$, and so on additional which is obtained depending upon the length of

the message.

Step 8: Alice chooses a private key from the generalized form of Fibonacci Q-matrix, she selects

$Q_p^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$ where $n = 0, \pm 1, \pm 2, \dots$ and $p=1$.

Step 9: Alice computes

$$p_1 = m_1 \times Q^n = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \times \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} q_1 & q_2 \\ q_3 & q_4 \end{pmatrix},$$

$$p_2 = m_2 \times Q^n = \begin{pmatrix} a_5 & a_6 \\ a_7 & a_8 \end{pmatrix} \times \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} q_5 & q_6 \\ q_7 & q_8 \end{pmatrix}, \text{ and so on.}$$

Step 10: The resultant set of points are

$T = \{q_1(x_1, y_1), q_2(x_2, y_2), q_3(x_3, y_3), \dots, q_i(x_i, y_i)\}$ where $i = 1, 2, 3, \dots$

Step 11: Compute $C_i = q_i + kabG$.

Step 12: Alice chooses a point R on the elliptic curve E , where k is the x-coordinate of R .

Step 13: Compute $R + abG$.

Step 14: Now Alice sends the encrypted message $(C_i, R + abG)$ to Bob.

3.2. Decryption

To recover the plain text q_i from C_i Bob do the following.

Step 1: First Bob selects the Alice public key $A = aG$.

Step 2: Bob uses his own private key 'b' and Computes $bA = b(aG)$.

Step 3: Now Bob computes the inverse element of $b(aG)$ is $-b(aG)$.

Step 4: Recover 'R' by adding $-b(aG)$ to the second part of the message: $-abG + R + abG = R$.

Step 5: Compute $k(abG)$ where k is the x-coordinate of R .

Step 6: Now Bob computes the inverse element of $k(abG)$: $-kabG$.

Step 7: Now Bob adds $-kabG$ to the first part of the message: $q_i + kbG - kbG = q_i$.

Step 8: After decryption, the obtained points are stored in 2×2 matrix as: