New Password Embedding Technique Using Elliptic Curve Over Finite Field

D. Sravana Kumar, C. H. Suneetha and P. Sirisha

Abstract In the present sophisticated digital era, safe communication of user password from one source to the other is quite difficult in client/server system. Also storing the password as it appears increases the potential risk of the security. Protection of the password is at most important in group communications to avoid the access of the illegal person to group resources. In addition, a roaming user who uses the network from different client terminals requires access to the private key. The present paper explains secure communication of password from one entity to the other. Here the password is encrypted using elliptic curve over finite field, embedded in a large random text at different selected positions, and communicated to the receiver via public channel.

Keywords Encryption · Decryption · Elliptic curve over finite field

1 Introduction

Since the digital network is growing explosively, the main difficulty in password communication mechanisms arises due to the fact that password is generally very small in length and simple. Exhaustive search attack will break the security of the password easily. Though the password is very small in size, it remains safe if it is communicated as cryptographic key. Diffie–Hellman key exchange protocol is an asymmetric cryptographic technique to negotiate session key over a secure channel which is revolutionary in the history of public key cryptography. But Diffie–Hellman

D. S. Kumar
Dr. V. S. Krishna Government Degree College, Visakhapatnam, India
e-mail: skdharanikota@gmail.com

C. H. Suneetha (✉)
GITAM University, Visakhapatnam, India
e-mail: gurukripachs@gmail.com

P. Sirisha
Faculty in Mathematics, Indian Maritime University, Visakhapatnam, India
e-mail: sirinivas06@gmail.com

N. R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, Advances in Intelligent Systems and Computing 906,
https://doi.org/10.1007/978-981-13-6001-5_15

key exchange protocol suffers from the defect of man-in-the-middle attack and does not establish the identity of the entity. Most of the modern research in cryptography is done in the direction of removing the weakness of Diffie–Hellman key exchange algorithm using public key certificates and digital signature.

2 Literature Survey

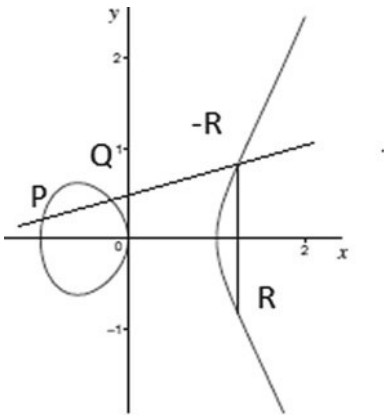
Several researches have been done pertaining to the attacks on key exchange protocols. Ku and Wang [1] established attacks on backward reply without modification and backward reply with modification. Aziz and Diffie [2] proposed an authentication protocol with requirement of pre-communication. This protocol uses cryptographic techniques of off-line certification procedures. But conventionally the cryptographic techniques need online certification of users with encrypted secret key. Later on, several modifications have been done by researchers to overcome the attacks on key exchange protocol. In this direction Authenticated Key Agreement (AK) protocols [3–7], Authenticated Key Agreement protocols with key confirmation [8–10] (AKC) came into picture. Perrig [11] proposed three different protocols suitable for applications with different types of security levels. Moreover, he intensified the strength of the efficiency of existing key agreement protocols using binary tree. In that chapter, he emphasized that these protocols reduce the number of encryption rounds from n to log (n) and reduce the bandwidth requirements from quadratic number to linear number. Key agreement protocols are contributory by all the individuals in the group. Burmster and Desmedt [12] suggested collaborative group key agreement protocols. They describe star-based, tree-based broadcast and cyclic protocols. But, all these protocols are only variations on Diffie–Hellman key exchange protocols. They have not addressed the problems of dynamic groups. Aydos et al. [3] proposed key agreement protocols for wireless communications using Elliptic Curve Cryptographic (ECC) techniques. Key agreement protocols basing on ECC produce considerable development to protect the integrity and confidentiality of the data like RSA and DSA.

3 Elliptic Curve Cryptography (ECC)

An affine equation E: $y^2 + b_1xy + b_3y = x^3 + b_2x^2 + b_4x + b_6$ over the set of real numbers is said to be Weierstrass equation, where b_1, b_2, b_3, b_4, b_6 and x, y are real numbers. An elliptic curve for cryptographic purpose is defined by the equation $y^2 = x^3 + ax + b$, $4a^3 + 27b^2 \neq 0$ over the finite field F_q.

Group laws of elliptic curve: Let E be an elliptic curve defined over the finite field of integers K. Addition of two points uses chord-and-tangent rule to get the third point [13–15]. The set of all points on the elliptic curve over the finite field with

Fig. 1 Geometric addition



addition as binary operation forms an abelian group with ∞, the point at infinity as identity element.

Geometric rules of Addition: Let $P(x_1, y_1)$ and $Q(x_2, y_2)$ be two points on the elliptic curve E. The sum of the two points P and Q is $R(x_3, y_3)$ which is the reflection of the point of intersection of the line through the points P, Q and the elliptic curve about x-axis. The same geometric interpretation also applies to two points P and $-P$, with the same x-coordinate. Here the points are joined by a vertical line, which is considered as the intersecting point on the curve at the point infinity. $P + (-P) = \infty$, the identity element which is the point at infinity [13–15].

3.1 Doubling the Point on the Elliptic Curve

If $P(x_1, y_1)$ is point on the elliptic curve, then $2P$ is the reflection of the point of intersection of the tangent line at P and the elliptic curve about x-axis [13–15]. Example of addition of two points and doubling of a point are shown in the following Figs. 1 and 2 for the elliptic curve $\boldsymbol{y^2 = x^3 - x}$.

Identity: $P + \infty = \infty + P = P$ for all E, where ∞ is the point at infinity [13–15].

Negatives: If $P(x, y)$ is a point on the elliptic curve, then $(x, y) + (x, -y) = \infty$, where $(x, -y)$ is the negative of P denoted by $-P$ [13–15].

Point Addition: If $P(x_1, y_1)$, $Q(x_2, y_2)$ are two points where $P \neq Q$, then $P + Q = (x_3, y_3)$ [13, 14], where $x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$ and $y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$.

Point Doubling: Let $P(x_1, y_1) \in E(\mathrm{K})$, where $P \neq -P$, then, $2\mathrm{P} = (x_3, y_3)$ [13, 14] where $x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1$ and $y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1$.

Point Multiplication: Let P be any point on the elliptic curve over finite field of integers. Then the operation multiplication of P is defined as repeated addition [16].

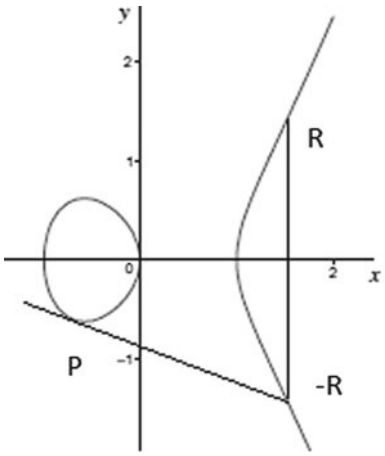


Fig. 2 Geometric doubling

$$kP = P + P + \cdots \text{ktimes}$$

3.2 Elliptic Curve Discrete Logarithmic Problem (ECDLP)

The robustness of ECC relies on the solution of Elliptic Curve Discrete Logarithmic Problem [16–18]. For two points $P, Q \in E_q(a, b)$, the elliptic curve over the finite field, it is hard to solve $x = \log_p Q$ for given $Q = xP$, where $x \in \{1, 2, \ldots, q - 1\}$.

4 Proposed Method

In view of the above survey, here authors designed secure password transmission protocol to overcome all the difficulties of key communication protocols. Here the password characters are encrypted embedded at different selected positions of a large random text using elliptic curve over finite field and transmitted to the intended recipient through public channel. This method provides the confidentiality of the password as well as the facility to store the password without any threat of stealing and can be retrieved whenever it is forgotten.

If Alice and Bob want to communicate with each other before communicating the messages, Alice selects an elliptic over finite field $E_q(a, b)$ having large number of points on it. In selecting the elliptic curve Alice takes precautions that the elliptic curve is secure for cryptographic purpose to escape from main physical attacks like side channel analysis and fault analysis. The desirable features of elliptic curves for encryption purpose to avoid the risk of attacks are 1. The order of the curve $\# E_q(a, b)$ should not be factorized into small primes to defend against Pohlig–Hellman attack

2. The curve should be non-super singular curve 3. It should be non-anomalous curve, i.e., the order # $E_q(a, b) \neq q$. In addition, Alice selects the elliptic curve $E_q(a, b)$ such that the order of the elliptic curve # $E_q(a, b)$ is a prime number because if the order of the group is prime, it is cyclic group. Alice chooses a generator $G(x, y)$ on the elliptic curve and also selects a random function $f(x, y, n)$ of the co-ordinates of generator $G(x, y)$ and a large integer n less than the order of the generator. She publishes the elliptic curve $E_q(a, b)$ and the function $f(x, y, n)$ on public channel. She shares the generator $G(x, y)$ with Bob as the secret key specific to Bob only for their communication. Again she selects a large random text having all types of characters (text/numerical/special) as medium for inserting the encrypted password. Since all the points on the elliptic curve are generators of the cyclic group of finite order, she shares another generator $G_1(x_1, y_1)$ with other communicating party Charles, $G_2(x_2, y_2)$ with Dickens, and so on.

Encryption:

If Alice wants to communicate the password having the characters or numbers $M_1M_2M_3M_4$.

1. She selects large random numbers $n, n_1, n_2, n_3, n_4, n_5$ less than the order of the generator and calculates the points $P = nG, P_1 = n_1G, P_2 = n_2G, P_3 = n_3G, P_4 = n_4G, P_5 = n_5G$.
2. Again she calculates $m_1 = [f(x, y, n)]_{\text{mod}n1}$, $m_2 = [f(x, y, n)]_{\text{mod}n2}$, $m_3 = [f(x, y, n)]_{\text{mod}n3}$, and $m_4 = [f(x, y, n)]_{\text{mod}n4}$. In computing these values, Alice takes care that no two m_j are equal for $j = 1, \ldots, 4$.
3. Alice encrypts the password $M_1M_2M_3M_4$ using logical XOR operation. She performs logical XOR operation between ASCII binary equivalents K_1, K_2, K_3, K_4 of M_1, M_2, M_3, M_4 and 8421 BCD codes of $[(n_5 + n_1)]_{\text{mod}100}$, $[(n_5 + n_2)]_{\text{mod}100}$, $[(n_5 + n_3)]_{\text{mod}100}$, $[(n_5 + n_4)]_{\text{mod}100}$ to obtain the 8-bit binary numbers K_{1E}, K_{2E}, K_{3E}, and K_{4E}.

$$K_{1E} = K_1 \text{ XOR}[(n_5 + n_1)]_{\text{ mod } 100}, K_{2E} = K_2 \text{ XOR}[(n_5 + n_2)]_{\text{ mod } 100}$$

$$K_{3E} = K_3 \text{ XOR}[(n_5 + n_3)]_{\text{ mod } 100}, K_{4E} = K_4 \text{ XOR}[(n_5 + n_4)]_{\text{ mod } 100}$$

Here mod100 is considered for performing logical XOR operation between 8-bit binary numbers K_1, K_2, K_3, K_4 and 8421 BCD codes of $[(n_5 + n_1)]_{\text{mod}100}$, $[(n_5 + n_2)]_{\text{mod}100}$, $[(n_5 + n_3)]_{\text{mod}100}$, $[(n_5 + n_4)]_{\text{mod}100}$.
4. The binary numbers K_{1E}, K_{2E}, K_{3E}, and K_{4E} are coded to characters using ASCII code table to obtain the cipher characters C_1, C_2, C_3, and C_4 of the password to be transmitted. These characters C_1, C_2, C_3, and C_4 are inserted at the positions m_1, m_2, m_3, and m_4 respectively, in the selected random text. For example, if $m_1 = 1256$, then the 1256th character of the random text is replaced by the first character C_1. Similarly, the character at m_jth position is replaced by C_j $j = 1, 2, 3, 4$.

5. Alice communicates the points $P, P_1, P_2, P_3, P_4, P_5$ and the random text in which the encrypted password characters are embedded to Bob via universal channel.

Decryption: Before decrypting the password, Bob constructs table of all points on the elliptic curve as multiples of generator G. After receiving the random text and points P, P_1, P_2, P_3, P_4, P_5 first picks up the values n, n_1, n_2, n_3, n_4, n_5 from the constructed table. $P = nG$, $P_1 = n_1G$, $P_2 = n_2G$, $P_3 = n_3G$, $P_4 = n_4G$, $P_5 = n_5G$.

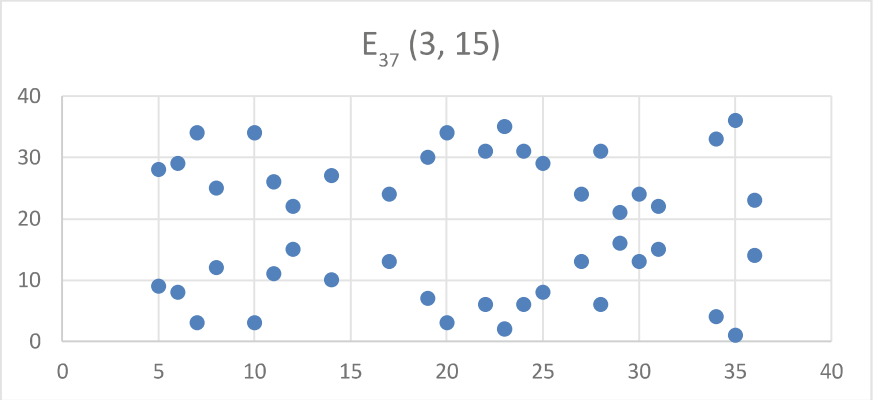
1. He calculates the function $f(x, y, n)$ and $m_1 = [f(x, y, n)]_{\mathrm{mod}n1}$, $m_2 = [f(x, y, n)]_{\mathrm{mod}n2}$, $m_3 = [f(x, y, n)]_{\mathrm{mod}n3,}$ and $m_4 = [f(x, y, n)]_{\mathrm{mod}n4}$.
2. Then Bob locates the encrypted password characters C_1, C_2, $C_{3,}$ and C_4 which are at the positions m_1, m_2, m_3, $m_{4,}$ respectively, in the received random text.
3. Bob writes equivalent 8-bit binary numbers for the characters of C_1, C_2, $C_{3,}$ and C_4 using ASCII code table to get K_{1E}, K_{2E}, K_{3E}, $K_{4E.}$ He decrypts K_{1E}, K_{2E}, K_{3E}, K_{4E} by executing logical XOR operation between K_{1E}, K_{2E}, K_{3E}, K_{4E} and 8421 BCD codes of $[(n_5 + \mathrm{n}_1)]_{\mathrm{mod}100}$, $[(n_5 + n_2)]_{\mathrm{mod}100}$, $[(n_5 + n_3)]_{\mathrm{mod}100}$, $[(n_5 + n_4)]_{\mathrm{mod}100}$ to obtain the binary numbers $K_1K_2K_3K_4$.

$$K_1 = K_{1E}\ \mathrm{XOR}[(n_5 + n_1)]_{\ \mathrm{mod}\ 100},\ K_2 = K_{2E}\ \mathrm{XOR}[(n_5 + n_2)]_{\ \mathrm{mod}\ 100}$$

$$K_3 = K_{3E}\ \mathrm{XOR}[(n_5 + n_3)]_{\ \mathrm{mod}\ 100},\ K_4 = K_{4E}\ \mathrm{XOR}[(n_5 + n_4)]_{\ \mathrm{mod}\ 100}$$

4. Corresponding ASCII characters of K_1, K_2, K_3, K_4 is the password sent by Alice.

Example: Consider an elliptic curve $y^2 = x^3 + 3x + 15$ and let $q = 37$. This curve satisfies all the above-listed required properties. Since the order of the curve $E_{37}(3, 15)$ is a prime number 47, the group is cyclic. The graph of the curve is,



Let the generator be $G(19, 30)$. The table expressing all the points as the multiples of generator is

$1G = (19, 30)$; $2G = (11, 26)$; $3G = (35, 36)$; $4G = (8, 25)$; $5G = (31, 15)$; $6G = (14, 10)$;

$7G = (20, 3)$; $8G = (24, 31)$; $9G = (34, 4)$; $10G = (12, 22)$; $11G = (36, 14)$; $12G = (30, 13)$;

$13G = (28, 31)$; $14G = (6, 29)$; $15G = (5, 28)$; $16G = (10, 3)$; $17G = (17, 13)$; $18G = (27, 13)$;

$19G = (25, 29)$; $20G = (29, 21)$; $21G = (22, 6)$; $22G = (23, 2)$; $23G = (7, 34)$; $24G = (7, 3)$;

$25G = (23, 35)$; $26G = (22, 31)$; $27G = (29, 16)$; $28G = (25, 8)$; $29G = (27, 24)$; $30G = (17, 24)$;

$31G = (10, 34)$; $32G = (5, 9)$; $33G = (6, 8)$; $34G = (28, 6)$; $35G = (30, 24)$; $36G = (36, 23)$;

$37G = (12, 15)$; $38G = (34, 33)$; $39G = (24, 6)$; $40G = (20, 34)$; $41G = (14, 27)$; $42G = (31, 22)$;

$43G = (8, 12)$; $44G = (35, 1)$; $45G = (11, 11)$ $46G = (19, 7)$; $47G = \infty$.

Encryption:

1. Let the function $f(x, y, n)$ selected by Alice be $f(x, y, n) = (8x + y)^n$. Alice publishes the elliptic curve $E_{37}(3, 15)$ and the function $f(x, y, n) = (8x + y)^n$ on public channel and shares the generator $G(19, 30)$ as the secret key with Bob. Let $n = 3$, $n_1 = 9$, $n_2 = 11$, $n_3 = 17$, $n_4 = 19$, $n_5 = 16$.
2. Alice calculates the points $P = nG = (35, 36)$, $P_1 = n_1G = (34, 4)$, $P_2 = n_2G = (36, 14)$, $P_3 = n_3G = (17, 13)$, $P_4 = n_4G = (25, 29)$, $P_5 = n_5G = (10, 3)$ and the function $f(x, y, n) = (8x + y)^n = f(19, 30, 3) = 6, 028, 568$.
3. Again she calculates $m_1 = (6{,}028{,}568)_{\text{mod}9} = 8$, $m_2 = (6{,}028{,}568)_{\text{mod}11} = 7$, $m_3 = (6{,}028{,}568)_{\text{mod}17} = 11$, $m_4 = (6{,}028{,}568)_{\text{mod}19} = 1$.
4. Let the password to be communicated is 'SSSS'. These characters are encrypted by performing logical XOR operation between ASCII binary equivalent of S and 8421 BCD codes of $[(n_5 + n_1)]_{\text{mod}100}$, $[(n_5 + n_2]_{\text{mod}100}$, $[(n_5 + n_3)]_{\text{mod}100}$, $[(n_5 + n_4)]_{\text{mod}100}$ to get 8 bit binary numbers K_{1E}, K_{2E}, K_{3E}, K_{4E}. These numbers are coded to corresponding ASCII characters to get encrypted password characters $C1$, $C2$, $C3$, and C_4.

$$K_{1E} = K_1 \,\text{XOR}[(n_5 + n_1)]_{\text{mod } 100},\ K_{2E} = K_2 \,\text{XOR}[(n_5 + n_2)]_{\text{mod } 100}$$

$$K_{3E} = K_3 \,\text{XOR}[(n_5 + n_3)]_{\text{mod } 100},\ K_{4E} = K_4 \,\text{XOR}[(n_5 + n_4)]_{\text{mod } 100}$$

The encrypted password for 'SSSS' is 'JHrp'. Alice selects a random plain text vachk#momn8pq125jabc#46)$WRU. The encrypted characters 'JHrp' are inserted at the positions 8, 7, 11, 1 of the random text. Then Alice communicates the random text 'pachk#HJmnrpq125jabc#46)$WRU' in which the encrypted password is embedded at the positions 8, 7, 11, 1 and the points (35, 36), (34, 4), (36, 14), (17, 13), (25, 29), (10, 3) to Bob in public channel.

Decryption:

1. Bob after receiving plain text 'pachk#HJmnrpq125jabc#46)$WRU' in which the encrypted password is embedded and the points (35, 36), (34, 4), (36, 14), (17,

13), (25, 29), (10, 3) first picks up the values $n = 3$, $n_1 = 9$, $n_2 = 11$, $n_3 = 17$, $n_4 = 19$, $n_5 = 16$ from the table of points he constructed as multiples of generator.
2. He calculates $f(x, y, n) = (8x + y)^n = f(19, 30, 3) = 6,028,568$, the values $m_1 = (6{,}028{,}568)_{\text{mod}9} = 8$, $m_2 = (6{,}028{,}568)_{\text{mod}11} = 7$, $m_3 = (6{,}028{,}568)_{\text{mod}17} = 11$, $m_4 = (6{,}028{,}568)_{\text{mod}19} = 1$ where the encrypted password characters are inserted.
3. Bob locates the encrypted password characters 'JHrp' in the received random text.
4. He performs logical XOR operation between ASCII binary equivalents K_{1E}, K_{2E}, K_{3E}, K_{4E} of the characters J,H,r,p and 8421 BCD codes of $[(n_5 + n_1)]_{\text{mod}100}$, $[(n_5 + n_2]_{\text{mod}100}$, $[(n_5 + n_3)]_{\text{mod}100}$, $[(n_5 + n_4)]_{\text{mod}100}$ to get 8-bit binary numbers K_1, K_2, K_3, and K_4.

$$K_1 = K_{1E}\ \text{XOR}[(n_5 + n_1)]_{\text{mod }100},\ K_2 = K_{2E}\ \text{XOR}[(n_5 + n_2)]_{\text{mod }100}$$

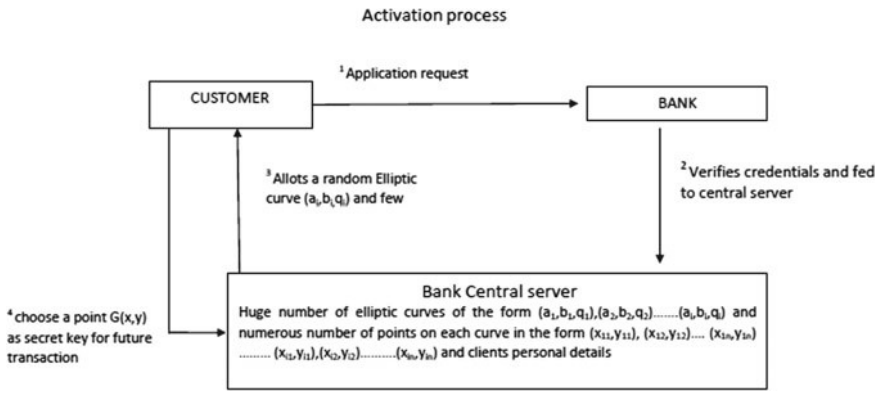
$$K_3 = K_{3E}\ \text{XOR}[(n_5 + n_3)]_{\text{mod }100},\ K_4 = K_{4E}\ \text{XOR}[(n_5 + n_4)]_{\text{mod }100}$$

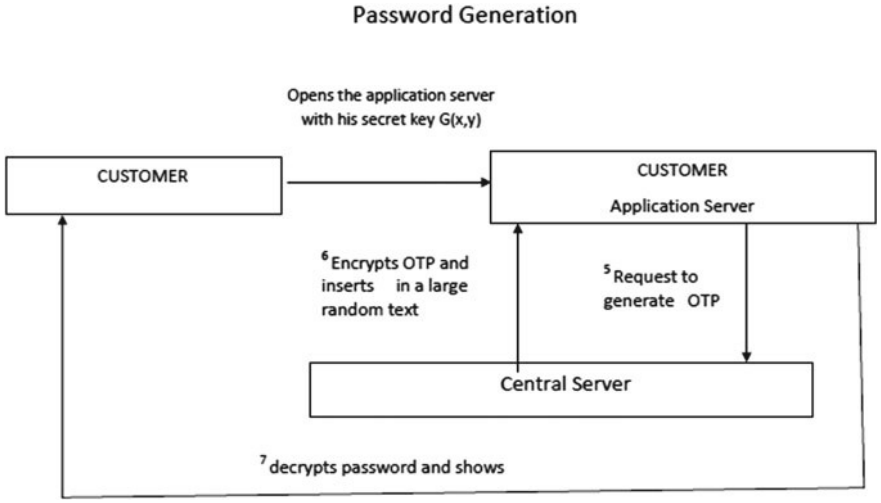
The equivalent ASCII characters of K_1, K_2, K_3, and K_4 reveal the password 'SSSS'.

5 Usage of the Protocol in Client/Server System

In the present paper, encrypted password embedding technique using elliptic curve over finite field is narrated. The password characters are encrypted and embedded in a large random text. The elliptic curve $E_q(a, b)$ and the function $f(x, y, n)$ are public. The encryption algorithm proposed here is more appropriate in client/server systems to send one-time password (OTP) to the clients for their transactions by central servers of banks or corporate. Initially, the communicating parties agree upon to use a point $G(x, y)$, the generator of $E_q(a, b)$ for transmission of different passwords for different transactions at different times. Here $G(x, y)$ is vital in this technique that acts as the secret key for their communication. If the elliptic curve is selected so that it has huge number of points on it, then every point is the generator of the cyclic group and acts as secret key. So, multiple numbers of persons can communicate with one another simultaneously. This provides ease and secure communication among multiple peers over universal channel. This is the reason that the password-embedded technique designed here is well suited for transactions in client/server system. As the central or bank server is powerful, it provides the information to workstations or clients. Enormous number of elliptic curves over finite fields in the form (a_1, b_1, q_1), (a_2, b_2, q_2), …, (a_i, b_i, q_i) having large number of points on each curve in the form (x_{11}, y_{11}), (x_{12}, y_{12}), …, (x_{1n}, y_{1n}); (x_{21}, y_{21}), (x_{22}, y_{22}), …, (x_{2n}, y_{2n}), …, (x_{i1}, y_{i1}), (x_{i2}, y_{i2}), …, (x_{in}, y_{in}) $[i = 1, 2, 3, \ldots]$ are stored in the central server. The client first approaches the bank for registration, and then the client requests for the

activation of the client's application server. At that stage, the central server allots an elliptic curve $E_q(a, b)$ at random and a few points on that curve to the client for choosing a point $G(x, y)$ as the secret key for future communication with the bank. Then central server stores the client's details and the secret key $G(x, y)$ chosen by the client in it. This process is called activation of the client's application. When the client needs transaction OTP, he/she request the central server to generate password. Then central server confirms the client's details, encrypts the OTP inserts in a large random text, and communicates to the client through public channel. Client server does the decryption procedure and shows the OTP to the client which is valid for a few seconds. As the central server is well built, many numbers of elliptic curves with large number of points on each curve are fed to it. Different curves with different points as secret keys can be allotted to several clients. So, various transactions from different clients can be performed by the central server simultaneously without overlapping.





6 Conclusions and Cryptanalysis

The main difficulty in password communication mechanisms arises due to the fact that length of the password is small and much easier to crack. Here in the present algorithm, the password characters are encrypted using elliptic curve over finite field and inserted at different selected positions of a large random text. Elliptic curve cryptography is more secure than classical methods by the strength of hard problem Elliptic Curve Discrete Logarithm Problem (ECDLP). The well-known attack to solve ECDLP is Pollard's Rho attack, i.e., finding $x = \log_P Q$ for $Q = xP$ where P, Q are points on elliptic curve using random walk technique and collision detection algorithm. Here the elliptic curve used and the function $f(x, y, n)$ are public, and the generator point $G(x, y)$ is secret or private key between two individuals. Sender transmits the large random text in which the encrypted key characters are inserted and the points $P, P_1, P_2, P_3, P_4, P_5$ through public channel. Since the generator G is secret and $n, n_1, n_2, n_3, n_4, n_5$ values are calculated by the receiver, the context of Pollard's Rho attack does not arise here. The password embedding technique presented here achieves all the required qualities of standard cryptographic algorithms.

References

1. Ku, W. C., & Wang, S. D. (2000). Cryptanalysis of modified authenticated key agreement protocol. *Electronic Letters, 36*(21), 1770–1771.
2. Aziz, A., & Diffie, W. (1994). A secure communications protocol to prevent unauthorized access: Privacy and authentication for wireless local area networks. In *IEEE Personal Communications*, pp. 25–31, first quarter.
3. Aydos, M., Sunar, B., & Koc, C. K. (1998). An elliptic curve cryptography based on authentication and key agreement protocol for wireless communications. http://www.researchgate.net.
4. Lee, C.-Y., Wang, Z.-H., Harn, L., & Chang, C.-C. (2011). Secure key transfer protocol based on secret sharing for group communications. *IEICE Transactions, 94-D*(11), 2069–2076.
5. Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory, IT-22*(6), 644–654.
6. Washington, L. C. (2008). *Elliptic curves: Number theory and cryptography* (2nd ed.). Boca Raton, FL: Chapman and Hall.
7. Baalghusun, A. O., Abusalem, O. F., Al Abbas, Z. A., & Kar, J. (2015). Authenticated key agreement protocols: A comparative study. *Journal of Information Security, 6,* 51–58.
8. Juels, A., Molnar, D., & Wagner, D. (2005). Security and privacy issues in E-passports. In *IEEE SecureComm'05*, pp. 74–88.
9. Black, U. (2009). "Other key security protocols" book. Teach yourself networking in 24 hours, 332p.
10. Bellare, M., Kilian, J., & Rogaway, P. (2000). The Security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences, 61,* 362–399.
11. Perrig, A. (1999). Efficient collaborative key management protocols for secure autonomous group communications. http://semanticscholar.org/18a4/025717de52e3981d67dd710a05ba2c926d2.pdf.
12. Burmster, M., & Desmedt, V.O. (1994). A secure and efficient conference key distribution system. In A. De Santis (Ed.), *EUROCRYPT 94*, LNCS 950, pp. 275–286.

13. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics Computation, 48*(177), 203–209.
14. Miller, V. (1985). Uses of elliptic curves in cryptography. In *Advances in Cryptography (CRYPTO 1985)*, Springer LNCS, Vol. 218, pp. 417–426.
15. Maurer, U., Menzes, A., & Teske, E. (2002). Analysis of GHS weil decent attack on the ECDLP over characteristic two fields of composite degree. *LMS Journal of Computation and Mathematics, 5,* 127–174.
16. Menzes, A., & Vanstone, S. (1997). Hand book of applied cryptography. In *The CRC-Press Series of Discrete Mathematics and its Applications*. CRC-Press.
17. Blumenfeld, A. (2011). *Discrete logarithms on elliptic curves*.
18. Miyaji, N., & Takano, S. (2006). Elliptic curves with low embedding degree. *Journal of Cryptology, 19*(4), 553–562.