DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5304

ISSN No. 0976-5697

Volume 9, No. 1, January-February 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

CRYPTOGRAPHIC KEY COMMUNICATION PROTOCOL USING ELLIPTIC CURVE OVER FINITE FIELD WITH NESTED CONSTRUCTION OF HASH MESSAGE AUTHENTICATION CODE

CH. Suneetha
Associate Professor
of Mathematics GITAM University
Visakhapatnam India

D.Sravana Kumar
Associate Professor
of Physics, Dr. VS Krishna Govt. Degree
CollegeVishakhapatnam India

P. Sirisha
Faculty in Mathematics
Indian Maritime University
Visakhapatnam India

Abstract : As the internet is the basic means of communication nowadays confirming the integrity and authenticity of the received data is a prime necessity in the communication network. The present paper explores the key communication protocol from one entity to the other rather than sharing the part of the key by both the entities using elliptic curve over the finite field. In addition, one Nested construction of Hash Message Authentication Code (NMAC) is tagged to the communicated key that guards the communicated key for verification using the cryptographic hash function. The protocol designed here is more suitable for communicating high secured passwords and PIN numbers in monetary transactions.

***KeyWords**:* Encryption, Decryption, Elliptic curve over finite field, NMAC

1. INTRODUCTION

Sometimes lightweight communications offer a high level of security, confidentiality, and authentication such as e-commerce and digital currency. In communications involving passwords, PIN numbers given by corporate and banks to the customers the input is very small in size. But the high level of security is expected even at the cost of computational risk and communication risk. Many password authentication protocols were designed by several cryptographers closely related to Diffie-Hellman key exchange protocol which is a revolutionary in the history of public key cryptography. But, this protocol is insecure against man -in- middle attack. Here both uses have no control over the secret key for their communication. Later, many authors published modifications to Diffie- Hellman key exchange algorithm over the years as Authenticated Key Agreement (AK) protocol and Authenticated key Agreement protocol with key confirmation (AKC) [5]

1.1 Authenticated Key Agreement (AK) protocol

In Authenticated key Agreement (AK) protocol between the legitimate pair of entities in the digital communication system, each entity has static or long-term public keys and the short-term public key called session keys.

- Consider two entities A and B in a group each having their own public keys, private keys, and the global parameters. Suppose that the entity A has two long-term public keys A1, A 2 and the other entity B has two long-term public keys B1, B2. The Authenticated Key Agreement protocol comprises the following steps:

• Entity A selects a random private session key, computes short-term public key or session key and forwards to B
• Entity B computes a long-term shared key K derived from A's public keys and B's private keys with some first function F_1
• By utilizing this long-term shared key K, A's public session key, B's Public session key
B computes an authenticated message and forwards to A
• A verifies the received authenticated message.
• Similar procedure is applied at the receiver's end.
• Confirmation of the secret between two members of the group is known as Authenticated Key agreement Protocol with key confirmation (AKC). i.e., one member satisfies the other member is really processing the key or not [10].

1.2 Desirable Features of Authenticated Key Agreement Protocols

- **Known Key Security**: The active participating entities should provide absolute secret key or session key which is not influenced even when the adversary acquires something about the other keys.

- **Forward Secrecy**: Previously designed session keys should not be infected by the
disclosure of long-term private keys of one or more entities.

- **Key Compromise Impersonation Resilience**: If the long term private key is revealed an adversary can involve in impersonation of the entity A (say) . But, this should not give the opportunity for the adversary to further impersonate other entities to A.

- **Unknown Key-Share Resilience**: The entities should be legitimate. i.e., Journal says A should not involve in malpractice in sharing the key with unauthorized entities [4].
- **Key Control**: The members involving in sharing process of the secret key do not have the knowledge of the key that is being established [4].

1.3 Disadvantages of Authenticated Key Agreement Protocol

The Authentic Key Agreement (AK) protocol and Authentic Key Agreement protocol with key confirmation (AKC) so far established are vulnerable against the above-listed attacks and security implications [3]. There is every possibility for the intruder to eavesdrops the data and modifies it even the legal entities have no control over what will be the key that is being established. Since each entity of the group contributes only the part of the key the entity has no knowledge about the rest of the part. The adversary can reflect the message to the same entity where the entity feels as if the message comes from the other side. In addition, the intruder can exhaust the resources sending a large number of similar protocols and even stops further establishing the key. So, Authentic Key Agreement protocols (AK) and AKC protocols have not attained the required level of security [4].
In the present paper, we propose an innovative key communication protocol using elliptic curve over finite which achieves all the desirable security attributes AK and AKC and overcomes the weaknesses of the key agreement protocols. Besides a Nested form of Hash Message Authentication Code (NMAC) is tagged to the communicated key i.e., the message the using the cryptographic hash function to provide additional integrity and authenticity. This tag is recomputed and verified at the receiver's end.

2. ELLIPTIC CURVE CRYPTOGRAPHY

2.1 Elliptic Curve Arithmetic:

A set of points satisfying the Weirstrass equation $y^2 + axy + by = x^3 + cx^2 + dx + e$ is called an Elliptic curve.

Here the variables x,y and the constants a,b,c,d,e are field elements, the point 0 is the point at infinity or zero point. Under addition as binary operation the set of all points on the elliptic curve forms an abelian group with identity element 0.
Point addition: Let $P(x_1,y_1)$, $Q(x_2,y_2)$ E (K) where P, Q
Then $P+Q=(x_3,y_3)$ [11].

$$x_3=\left(\frac{y_2-y_1}{x_2-x_1}\right)^2 - x_1-x_2$$

$$y_3=\left(\frac{y_2-y_1}{x_2-x_1}\right) -(x_1-x_2)-y_1$$

Point Doubling: [11] Let $P=(x_1,y_1)$ be a point on the elliptic curve $P\neq -P$, $2P=(x_3,y_3)$ where

$$x_3=\left(\frac{3x_1^2+a}{2y_1}\right)^2 -2x_1$$

$$y_3=\left(\frac{3x_1^2+a}{2y_1}\right) (x_1-x_3)-y_1$$

2.2 Elliptic Curves defined over finite field

For Elliptic Curve Cryptography the curves are defined over some finite field $F_p(p = 23)$ where p is a large prime number. For the purpose of encryption and decryption using elliptic curves it is sufficient to consider the third order equation of the form $y^2 = x^3 + ax + b$ over the prime field F_p , where all the coefficients and the variables take all values in the set of integers from 0 through p-1 and the elliptic field arithmetic is done in this field F_pWith respect to modulo p.

2.3 Elliptic Curve Discrete logarithmic Problem (ECDLP):

Elliptic curve cryptography depends on the strength of hard problem called elliptic curve discrete logarithm problem (ECDLP).For any two points Q, P on Ep (a, b) consider an equation Q =kP where Q, P ∈ Ep (a, b) and KP. It is relatively easy to calculate Q given k and P. But it is relatively hard to determine k given Q and P [11].

3. HASH FUNCTION:

A function that takes variable-length input, transforms to a fixed length output is called Hash Function. The output is called hash value or message digest. A hash function used for security purpose is called cryptographic hash function h = H (M).
$H: D_v \rightarrow R_F$ where D_v is the domain having variable length input and R_F is the range having fixed length output.

3.1 Properties of Cryptographic Hash Functions

A good cryptographic hash function should have the following properties to resist all types of active and passive attacks [6,7]
• Pre-Image Resistant: For the given hash value it is difficult to find the message m such that h = hash (m).i.e., the function should be a one-way function.
• Second Pre-Image Resistant: For given hash value h it is infeasible to find a function y such that h = H(y) computationally.
• Collision Resistant: For two different messages m_1,m_2 it is impossible to find
$H(m_1) = H(m_2)$ [7,8]. In present days most widely used cryptographic hash functions are
MD5 and SHA - 1.

3.2 Applications of cryptographic hash functions

The cryptographic hash function is most adaptable algorithm having many applications. Generally cryptographic, the hash is divided into two types keyed-hash functions called Message Authentication Code and un keyed-hash functions. Keyed Hash function is used to verify the genuineness of the message. In this mechanism the hash code of the secret key is encrypted to achieve the required authentication. The message and the concatenated hash code of the secret key are encrypted to attain the required authentication as well as confidentiality [1, 2].
Hash Message Authentication Code (HMAC) is constructed

as

HMAC (K,M) = H [(K^+ ⊕opad) ||H[(K^+⊕ipad) ||M]] using hash functions like MD 5,

SHA – 1 etc [9]

Here K^+ is the key created by by padding extra zeros to the right on the right side of the input block. iPad means inner padding and opad means outer padding of the bits [10].

⊕represents exclusive OR operation.

Nested construction of Message Authentication Code (NMAC)

To construct NMAC two keys are used here. First the message is concatenated with one key and hash value is calculated. Then the output is concatenated with the second key, again the hash value is calculated [9]. For two keys K_1, K_2 $NMAC(K_1,K_2,M)=h_{K_1}(H_{K_2}(M))$

Digital Signature: In digital signature the hash value is encrypted using sender's private key. Here the security relies on the rigidity of finding the collision. To achieve both confidentiality and digital signature it is suggestible to encrypt the message and private-key encrypted hash code [9]. Moreover these applications cryptographic hash functions are used to learn interruption detection and virus detection also.

4. PROPOSED METHOD

In key agreement protocols, all the members of the group contribute the information to establish the shared secret key. The disadvantages of these protocols are discussed in section 2.3. If two legitimate entities Alice and Bob in the group of network want to communicate the messages through the public channel with absolute security, one of the means is using a one – time key [10]. Here we propose a new technique for key communication protocol where the secret key is constructed by one entity of the group and communicates to the other entity in a public channel network using the elliptic curve over finite field

Abbreviations used in this Key Communication Algorithm

E_P (a,b): Agreed upon Elliptic Curve by the entities in the group of network over the finite field p

C: Agreed upon point by the entities on the elliptic curve E_P (a,b)

λ: Alice's long-term or static private key 1, a large random number less than the order of E_p (a, b)

A: Alice's long-term or static private key 2, a point on E_p (a, b)

α: Alice's short-term or ephemeral private key, a large random number less than the order of E_p (a, b)

A_1: Alice's long-term or static public key 1, a point on E_p (a, b)

A_2: Alice's long-term or static public key 2, a point on E_p (a, b)

A_B: Alice's short-term or ephemeral public key, a point on E_P (a,b) specific to Bob only

μ: Bob's long-term or static private key 1, a large random number less than the order of E_p (a, b)

B: Bob's long-term or static private key 2, a point on E_p (a, b)

β: Bob's short-term or ephemeral private key, a large random number less than the order of E_p (a, b)

B_1: Bob's long-term or static public key 1, a point on E_p (a, b)

B_2: Bob's long-term or static public key 2, a point on E_p (a, b)

B_A: Bob's short-term or ephemeral public key, a point on E_P (a,b) specific to Alice only. Alice and Bob agree upon to use the elliptic curve E_p (a, b), and a point C on it. Alice selects a large random number λ , and a point A on the elliptic curve, Where the random number λ is less than the order of the generator of the elliptic curve Ep (a, b) She computes

$A_1 = \lambda(C + A)$ and

$A_2=\lambda C$

She keeps the random number λ and the point A as her secret keys and publishes A_1 and A_2 as her long-term or static public keys.

Similarly, Bob selects a large number μ and a point B on the elliptic curve. He computes

$B_1 = \mu(C + B)$ and

$B_2= \mu C$

He keeps the random number μ and the point B as his secret keys and publishes B_1 and B_2 as his long-term public keys or static public keys.

Again Alice chooses a random number less than the order of the generator say α and reconstructs $A_B=\lambda\alpha B_2$and publishes it as her short-term or ephemeral public key that is specified for Bob only to communicate with him.

In the same way, Bob chooses a random number β andreconstructs$B_A=\mu\beta A_2$and publishes it as his ephemeral or short-term public key specific for Alice only to communicate with her.

4.1 Secret key Encryption:

Assume that Bob wants to use a point S on the elliptic curve as the secret key when he wants to communicate with Alice. A pair of numbers will be given by secret key S. Bob encrypts the secret key S as follows.

$S^E = S + \beta A_B + \left(\frac{B_1}{\mu} - B\right)$

4.2 Construction of Nested Form of Hash Message Authenticated Code (NMAC)

Bob computes the Nested form of Hash Message Authentication Code (NMAC) using his long-term public keys B_1, B_2 and the secret key S^E to be transported (i.e., the message to be sent to the other entity). Here B_1, B_2, S^E are points on elliptic curve E_p (a, b).

Suppose $B_1=(b_1^1, b_1^2)$,$B_2=(b_2^1, b_2^2)$,$S_E=(S_1^E, S_2^E)$

First Bob calculates P_1, P_2 the points on the elliptic curve as

$P_1=B_1.S^E = (p_1^1, p_1^2)$

$P_2=B_2.S^E = (p_2^1, p_2^2)$

For conventional encryption Bob generates a single number from the pair of numbers i.e., the coordinates of the point on the elliptic curve using some function F(x, y), say $F(x,y)=x^2+y^2$.Bob computes the secret keys K_1, K_2 applying the function F(x,y) on the above-computed point s P_1, P_2 as

$K_1=F[P_1]=F[p_1^1, p_1^2]=[p_1^1]^2 + [p_1^2]^2$

K_2=F[P_2]=F[p_2^1, p_2^2]=$[p_2^1]^2 + [p_2^2]^2$
the message M=$(s_1^E)^2 + (s_2^E)^2$
Then the nested form Hash Message Authentication Code(NMAC) is calculated as

$$NMAC(K_1,K_2,M)=h_{K_1}(H_{K_2}(M))$$

Bob communicates the secret key S^E to Alice along with the nested form of Hash Message Authentication Code (NMAC). The scheme for construction of the keys K_1, K_2 in the nested form of Hash Message Authentication Code (NMAC), function F(x,y) and the hash algorithm used are the secret agreement between the active participating entities in the network. The hardness of the Nested form of Hash Message Authentication Code (NMAC) relies on the scheme of construction and the parameters involved. To defend the success probability of the adversary the function F(x,y) may be varied time to time, or the function F(x,y) may be defined as
F(x,y) = $x^2 + y^2 + N^2$ where N is the constant, a natural number 1,2,3,.......
i.e., N is a constant takes 1 for the first message, 2 for the second message and soon.

4.3 Decryption

Verification of Hash Message Authentication Code
Alice after receiving the secret key along with the Hash Message Authentication Code first computes the secret keys K_1, K_2 by using Bob's long-term public keys B_1, B_2, the communicated key S^E (i.e., the message M) and the agreed upon Function F(x,y). Then she verifies the calculated hash value is same as received hash value. After confirming the Nested form of Hash Message Authentication Code (NMAC) Alice decrypts the transported secret key as follows

Alice retrieves S as

$$S = S^E - \alpha B_A - \left(\frac{A_1}{\lambda} - A\right)$$

The Decryption works out properly:-

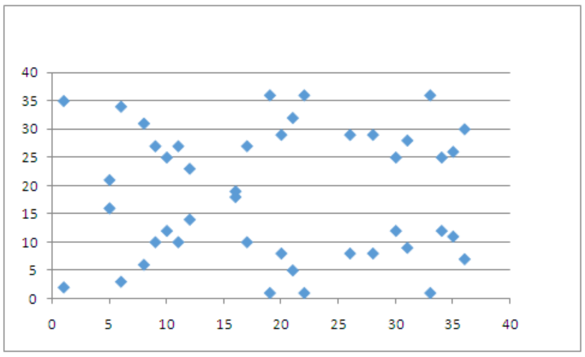
$$S^E = S + \beta A_B + \left(\frac{B_1}{\mu} - B\right)$$

$$S = S^E - \alpha B_A - \left(\frac{A_1}{\lambda} - A\right)$$

$$S + \alpha\beta\lambda\mu C + (B + C - B) - \alpha\beta\lambda\mu C - (A + C - A)$$
=S

5. EXAMPLE

Now consider the elliptic curve E $_{37}$(-5, 8).
The points on the curve E $_{37}$(-5, 8) are
{(1,2),(1,35),(5,21),(5,16),(6,3),(6,34),(8,6),(8,31),(9,27),(9,10),(10,25),(10,12),(11,27),(11,10),(12,23),(12,14),(16,19),(16,18),(17,27),(17,10),(19,1),(19,36),(20,8),(20,29),(21,5),(21,32),(22,1),(22,36),(26,8),(26,29),(28,8),(28,29),(30,25),(30,12),(31,9),(31,28),(33,1),(33,36),(34,25),(34,12),(35,2),(35,11),(36,7),(36,30),∞}
The graph of the function



Let C= (11, 10).A random λ=5 and a point A= (9, 27) on the elliptic curve has been selected by Alice and She computes
$A_1=\lambda(C + A)$=5[(11, 10) + (9, 27)] = (5, 16)
$A_2 = \lambda C$ = (30, 25)
She keeps the random number λ and the point A on the elliptic curve as her long-term or static private keys and publishes A_1 and A_2 as her long-term or static public keys.
Similarly Bob selects μ =8, B= (16, 18) on the elliptic curve. He computes
$B_1= \mu(C + B)$ =(30, 12)
$B_2=\mu C$= (1, 2).
He keeps the random number μ and the point B on the elliptic curve as his long-term or static private keys and publishes B_1 and B_2 as his long-term or static public keys.
Again Alice selects a random number α = 3, computes $A_B=\lambda\alpha B_2$= (12, 23) and publishes as her ephemeral or short-term public key specific to Bob.
Similarly, Bob selects $\beta = 6$ computes $B_A=\mu\beta A_2$= (26, 29) and publishes as his short time public key specific to Alice only.
Encryption of the secret key by Bob: If Bob wants to send the secret key S= (8, 31) on the elliptic curve E $_{37}$ (-5, 8) he encrypts
S^E=S+ $\beta A_B + \left(\frac{B_1}{\mu} - B\right)$= (33, 36).
Bob calculates the points P_1 and P_2 on the elliptic curve as
$P_1=B_1.S_E$= (20, 29). (24, 19)= (480,551) = (P_1^1, P_1^2)
$P_2=B_2.S_E$=(44,43).(24,19))=(1056,817)=(P_2^1, P_2^2)
For conventional encryption Bob generates a single number from the pair of numbers say
F(x,y)=x^2+y^2.
B_1=(30,12),B_2=(1,2),S^E=(33,36) are the points on the elliptic curve E_{37}(-5,8)
Bob calculates the points P_1 and P_2 on the elliptic curve as
$P_1=B_1.S^E$= (30, 12). (33, 36)= (30, 12) = (P_1^1, P_1^2)
$P_2=B_2.S^E$= (1, 2). (33, 36)= (33, 36) = (P_2^1, P_2^2)
For conventional encryption Bob generates a single number from the pair of numbers (x, y) say
F(x, y) = x^2+y^2.
Bob computes K_1, K_2 as
K_1=F [P_1]=$[30]^2+[12]^2$=900+144=1044
K_2=F[P_2]=$[33]^2+[36]^2$=1089+1296=2385
M= $(S_1^E)^2 + (S_2^E)^2$= $(33)^2+(36)^2$=1089+1296=2385

$$NMAC(K_1,K_2,M)=h_{K_1}(H_{K_2}(M)) =$$
367d0b50d04fcc7aa9280464d018371560864000 (SHA – 1)

Decryption: Alice after receiving the secret key and Nested form of Hash Message Authentication Code (NMAC) first verifies the NMAC. After confirming the NMAC she decrypts the secret key as

Alice retrieves S as S= $S^E - \alpha B_A - \left(\frac{A_1}{\lambda} - A\right)$ = (8, 31)

6. SECURITY ANALYSIS AND CONCLUSIONS:

The key communication protocol proposed here achieves all the desirable security attributes of Authentic Key Agreement (AK) protocol as well as non-repudiation of the communication among the entities of the group.

Known Key – Security:

In the key communication protocol designed here the ephemeral or short-term public key is published by each active entity in the group specific to the other. i.e., the entity A publishes public key A_B specific to B only to communicate with B using B's long-term or static public keys. In computing the ephemeral public key entity A uses a random number α, user B uses a random number β. The random numbers α, β are different for different ephemeral or short-term public keys. A and B discards the short term public key and reconstruct time to time. So, the proposed algorithm suggested defends against known – key security attack.

Forward Secrecy: Even if the static private keys of the entities are compromised the previous session keys established by legitimate entities are not influenced because the formation of ephemeral keys is protected by the hard problem called Elliptic Curve Discrete Logarithm Problem (ECDLP). In addition computation of ephemeral or short-term public keys involve the parameters α, β those differ for each communication. So, the present algorithm possesses best forward secrecy.

Key compromise impersonation If A's long-term or static private key is revealed the adversary can impersonate A to B. But the adversary cannot impersonate B to A unless he is aware of the knowledge about B's long-term private key. The construction of long term and short term public keys are protected by the hard problem Elliptic Curve Discrete Logarithmic Problem (ECDLP) So, the present algorithm is safe against the key compromise impersonation.

Unknown key – share The key communication protocol described here also prevents the unknown-key share because both the users use their own private keys, long-term public keys, and specific short-term public keys.

Besides all these security attributes the secret key communicated is shielded with the Nested form of Hash Message Authentication Code (NMAC). HMAC value is calculated by the sender utilizing his static or long-term public keys and the secret key which is being communicated called the message. In calculating the NMAC value the active entities in the group of network concur to use a random function F(x,y), a mechanism to construct the keys K_1, K_2, available Hash algorithms like MD 5, SHA – 1 etc. To conquer the active attacks on NMAC value the random function F(x,y) is changed periodically. The message to be communicated (i.e., the secret key S^E) is concatenated with the sender's long-term public keys in calculating the Nested form of Hash Message Authentication Code (HMAC) value. At the receiver's end, the receiver first calculates NMAC value and verifies the NMAC value with received hash value to authenticate the communication. The summary of this algorithm is it provides at most security in key communication that is guarded by the Nested form of Hash Message Authentication Code (NMAC).

REFERENCES

1. B. Preneel and P. Van Oorschot On the security of two MAC algorithms Advances in Cryptology EUROCRYPT 96, Proceedings Lecture Notes in Computer Science Vol. 1070 U. Maurer ed., Springer Verlag 1996.
2. A. Menzies, P. Van Oorschot and S. Vanstone Handbook of Applied Cryptography CRC Press 1997.
3. B. Song and K. Kim Two- pass Authenticated Key agreement protocol with key confirmation Progress in Cryptology INDOCRYPT LNCS 1977 Springer – Verlag pp 237-249, December 2000.
4. Simon Blake-Wilson, Don Johnson, Alfred Menezes "Key agreement protocols and their security analysis", IMA International Conference on Cryptography and Coding
5. ANSI X9.63-1997, *Elliptic Curve Key Agreement, and Key Transport Protocols*, October 1997, working draft.
6. B. den Boer, A. Bosselaeres, "Collisions for the Compression function of MD 5" Advances in Cryptology EUROCRYPT ' 93 proceedings Springer – Verlag 1994.
7. X.Y. wang, F.D. Guo, X.J. Lai, H.B. Yu "Collisions for Hash functions MD 4, MD 5 HAVEL -128 and RIPEMD, rump session of CRYPTO'04 E-print 2004
8. "Hash Functions in Cryptography" Master of Science thesis Joseph Sterling Grah submitted to University I Bergen June 2008.
9. Junko Nakajima and Mitsuru Matsui "Performance Analysis and Parallel Implementation of dedicated hash functions" Proc. Of EUROCRYPT 2002 Lecture Notes in Computer Science 2332 Springer pp165-180 2002
10. Narn-Yih Lee, Chein – Nan Wu, "Authenticated multiple key exchange protocols based on Elliptic curves and bilinear pairings" Computers & Electrical Engineering
    Volume 34, Issue 1, January 2008, Pages 12-20
11. Smart N.P. "The Discrete Logarithm on Problem on Elliptic Curve of Trace one" Journal of Cryptology 1999 Vol. 12, No.3, Page 193-196.