A public key data encryption based on elliptic curves

NISSA MEHIBEL and M'HAMED HAMADOUCHE
LIMOSE laboratory
Faculty of sciences
University M'hamed Bougara of Boumerdes, Independence Avenue-35000-Algeria.
nissa.mehibel@gmail.com hamadouche-mhamed@hotmail.com

Abstract: - The basic Elliptic Curve Cryptosystem used for encryption is Elgamal encryption scheme, it is a public key cryptosystem which is based on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP). However, this cryptosystem does not guarantee all the security constraints. In this article, we propose a new public key cryptosystem that provides all the security constraints such as confidentiality, integrity, authenticity and non-repudiation of the data using Elliptic Curves.

Key-Words: - ECC; ECDLP; Elliptic Curve Diffie-Hellman key exchange; Elgamal cryptosystem; public-key encryption; Hash function; man-in-middle attack.

1 Introduction

Asymmetric cryptography was introduced in 1976 by White Diffie and Martin Hellman [1] in order to solve the problem of secret key exchange in unsecured channels, their secret key exchange protocol was based on the discrete logarithm problem. The Discrete Logarithm problem for a general algebraic group G can be stated as follows: given $\alpha \in G$, find an integer such that $\beta=\alpha^x$, provided that an integer exists. The integer x is called Discrete Logarithm [2]. Diffie and Hellman have also proposed the theoretical model of an asymmetric cryptosystem, which consists of using two keys: a public key to encrypt, and a private key to decrypt data. This model has been applied by Elgamal [3].

Nowadays, there is a growing interest in Elliptic Curve Cryptography, which was developed in 1985 independently by two mathematicians, Neal Kobiltz [4] and Victor Miller [5]. It is a public key cryptography based on the arithmetic of elliptic curves and security of the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Elliptic Curve Cryptography can be used for encryption, digital signature and key exchange. The advantage of elliptic curves is that they provide a level of security equivalent to that of existing public key systems but with shorter key lengths [6].

The basic Elliptic Curve Cryptosystems such as Diffie-Hellman key exchange and Elgamal encryption scheme are vulnerable to man-in-the-middle attacks. There exists several research works that treated of man-in-middle attacks.

To ensure authentication in ECC, two methods exist, the first method consists of adding a digital signature scheme to the cryptosystem, while the second method consists in modifying the Diffie-Hellman key exchange such that the digital authentication is integrated into the exchange scheme [10].

In the first method, we find Elliptic Curve Digital Signature Algorithm ECDSA [11], which is the first algorithm of digital signature on EC. EDSA is the analogue of Digital Signature Algorithm DSA [12]. However, it is safer and faster than DSA. In 2011, two algorithms (ECDSA1 and ECDSA2) were proposed, which are an improvement of ECDSA. These algorithms reduce the computational cost while keeping the same security level as the original ECDSA [13]. In 2016, ECDSA1 and ECDSA2 were improved by using two random numbers of signature generation in order to reduce the probability of secret key risk exposure to potential attacks [14].

In the second method, there are two ways of ensuring the authentication of the key. The first approach is the authenticated key agreement, where we find several works that propose to add a digital certificate [15], [16], [17] or add hash functions [18], [19], [20]. The second approach is the authentic key transport between two communicating parties. Here, we find the work proposed in [21], where key authenticity is ensured by publishing the

specific public key for each communicator. In addition, this specific public key is used in [22], where an algorithm of asymmetric encryption is proposed to ensure data authenticity.

This paper is a revised and expanded version of the work presented in [23]. In this paper, we propose a new public-key encryption system that satisfies all the security constraints such as data confidentiality and integrity, users' authenticity, and transactions non-repudiation. Hence, the proposed cryptosystem is less susceptible to the man-in-middle attack. The rest of the paper is organized as follows. Section 2 gives basic mathematical notions on which Elliptic Curve Cryptography is based. Section 3 summarises Elliptic Curve Encryption. Section 4 briefly gives the concept of a hash function. Section 5 presents the proposed cryptosystem. In Sections 6, comparisons are done, and Section 7 concludes the paper.

2 Basic notions

The elliptic curve E over a finite field is an algebraic curve non-singular [24], which can be represented by the generalized Weierstrass equation:

$$E:\{(x,y)|y^2+\alpha_1xy+\alpha_3y-x^3-\alpha_2x^2-\alpha_4x-\alpha_6=0\}\cup\{O\} \quad (1)$$

Where α_1, α_2, α_3 ,α_4 and $\alpha_6 \in E$ and O the point at infinity.

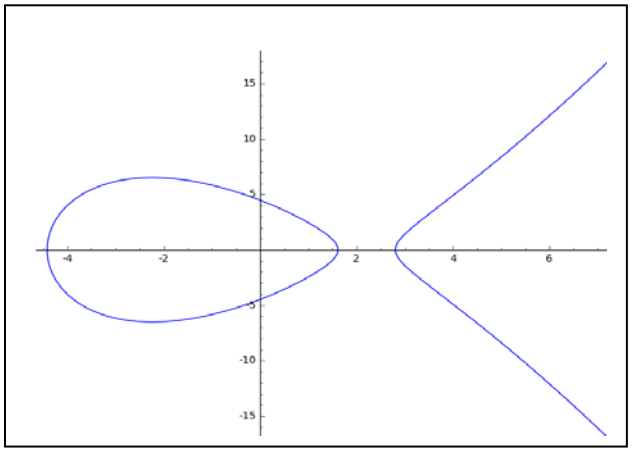

Fig. 1. Graph of Elliptic Curve $y^2=x^3-15x+20$.

In the present paper, we study an elliptic curve of third degree over a finite field $k = F_p$ having the form (2) shown in the Fig. 1. for the elliptic curve $y^2=x^3-15x+20$.

$$y^2=x^3+ax+b \quad (2)$$

Where $a,b \in k$ and $4a^3 + 27b^2 \neq 0$ .Together with an extra point O, called the point at infinity.

2.1 Group low of elliptic curve

Let $E_p(a,b)$ be an elliptic curve defined in a finite field $k = F_p$, the set of points on the curve with the point at the infinity, denoted ∞, form an Abelian group G whose composition law is the addition of points satisfying the following conditions:

- *Closure:* $\forall(P_1,P_2) \in E_p(a,b) \mid P_1 + P_2 \in E_p(a,b)$
- *Commutativity:* $\forall (P_1,P_2) \in E_p(a,b) \mid P_1 + P_2 = P_2 + P_1$
- *Associativity:* $\forall (P_1,P_2,P_3) \in E_p(a,b) \mid (P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$
- *Identity element* ∞: $\forall P \in E_p(a,b) \mid P + \infty = \infty + P =$
- *Symmetrical element*: $\forall P(x,y) \in E_p(a,b)$, $\exists Q(x,-y) \in E_p(a,b) \mid P + Q = Q + P = \infty$ (Q is a symmetrical element of P denoted by –P).

2.2 Geometric addition

Let $P(x_1,y_1)$ and $Q(x_2,y_2)$ be two points on the elliptic curve $E_p(a,b)$ [23,24]

- *Point adding:* $P \neq Q$, the group operator will allow us to calculate a third point $P + Q = R(x_3,y_3) \in E_p(a,b)$ where

  $x_3 = (y_2 - y_1 / x_2 - x_1)^2 - x_1 - x_2$ and

  $y_3 = (y_2 - y_1 / x_2 - x_1)(x_1 - x_3) - y_1$

- *Point doubling*: P = Q, the group operator will allow us to calculate a third point P + P = 2P = $R(x_3,y_3) \in E_p(a,b)$ where

  $x_3 = (3x_1^2 - a / 2y_1)^2 - 2x_1$ and

  $y_3 = (3x_1^2 - a / 2y_1)^2 - (x_1 - x_3) - y_1$

- *Adding vertical point:* P = -P, the group operator will give us the point at infinity $P + (-P) = \infty$.

2.3 Point multiplication

Let P(x,y) be any point on the elliptic curve $E_p(a,b)$ and k is a large integer scalar multiplication consisting of computing the value k*P by doing a series of point doublings and additions until the

product point is reached [23,24]. kP = P +P + P…+ P, k times.

2.4 Elliptic Curve Discrete Logarithm Problem

Elliptic Curve Cryptography (ECC) is based on the difficulty of Elliptic Curve Discrete Logarithmic Problem (ECDLP), the difficulty of this problem is to determine the value of k of the equation Q = k*P for the known points P and Q on the elliptic curve $E_p(a,b)$, where k is a Large random number less than p.

3 Elliptic Curve Encryption

The basic public-key cryptosystem proposed to encrypt and decrypt messages is the Elgamal cryptosystem [9]. When Alice wants to send a secret message to Bob, so she does the following [23].

1. Alice downloads Bob's public key Q_b , elliptic curve domain parameters $E_p(a,b)$ and point generator G, where $Q_b = n_b * G$ and n_b is a private key of Bob .

2. She transforms its message into a point $M \in E_p(a,b)$.

3. She chooses a secret integer k and computes $C_1 = k*G$.

4. She calculates $C_2 = M + k * Q_b$.

5. She sends the encrypted message $C_i = \{ C_1, C_2\}$ to Bob.

Bob decrypts a message as follow

1. Bob receives the encrypt message $C_i = \{C_1, C_2\}$.
2. He calculates $M_1 = n_b * C_1$.
3. He calculates $M = C_2 - M_1$.

The main weakness of the cryptosystem of Elgamel is that the authenticity of the message is unsure, an attacker can easily create a common key with the entities, so the entities think to communicate to each other, while they are actually communicating with the attacker [23].

4 Hash Function

A hash function or a digest function is a function that transforms a message M to a value h called hash, H (M) = h, it is used in cryptography to satisfy the following properties:

- In input, the size of message can be of any length, whereas in output, the h has a fixed value.
- The computation of H (M) can be calculated very quickly.
- H(M) is one way function (invertible), i.e. using the hash h, it is impossible to find the message M from h.
- H(M) is collision-free, given two messages M_1 and M_2, it is infeasible to find H(M_1) = H(M_2).

The hash functions can be used as a digital fingerprint for a message or to check the integrity of the data in order to verify if the message has been corrupted or not. Examples of well known hash functions are MD2, MD5 and SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 [25]. In the cryptosystem proposed in the next section, the SHA-1(M) function is used to hash the data.

5 Proposed cryptosystem

In this section, a new method to encrypt and decrypt messages ensuring the confidentiality, integrity, authenticity and non-repudiation of messages, is proposed. Our contribution consists in introducing the specific public key for each communicating party [23,24] and a hash function.

Two communicating parties Alice and Bob begin to generate their keys and then exchange them as in Algorithm 1 in accordance with the following steps:

- Alice and Bob agree upon to use an elliptic curve $E_p(a,b)$ where p is a prime number and a generator G of $E_p(a,b)$.
- Alice selects a large random number $n_a <$ ord(G) and a point P_a on the elliptic curve as her secret keys, and calculates $Q_a = n_a * G$ and publishes it as her public key.
- Bob selects a large random number $n_b <$ ord (G) and a point P_b on the elliptic curve as his secret keys and calculates $Q_b = n_b * G$ and publishes it as his public key.

- Alice calculates $S_a = n_a*(P_a + Q_b)$ and publishes it as her specific public key for Bob.
- Bob calculates $S_b = n_b*(P_b + Q_a)$ and publishes it as his specific public key for Alice.

Algorithm 1 key generation

Input : p, E and G
Output: the keys (n_i ,P_i , Q_i, S_i)

// **Generation of public key**

If (Alice)
1- Select n_a // $n_a <$ ord (G), private key
2- Select P_a // $P_a \epsilon$ E_p(a,b) , private key
3- Compute $Q_a = n_a * G$ // public key
4- Send (Q_a, Bob) // Send Q_a to Bob

If (Bob)
1- Select n_b // $n_b <$ ord (G), private key
5- Select P_b // $P_b \epsilon$ E_p(a,b) , private key
2- Compute $Q_b = n_b * G$ // public key
3- Send (Q_b, Alice) // Send Q_b to Alice

// **Generation of specific public key**

If (Alice)
1- Compute $S_a = n_a*(P_a + Q_b)$ // specific public key
2- Send (S_a, Bob) // Send S_a to Bob

If (Bob)
1- Compute $S_b = n_b*(P_b + Q_a)$ // specific public key
2- Send (S_b, Alice) // Send S_b to Alice

For encryption, Alice and Bob calculate and exchange their ciphertext following the steps in Algorithm 2. In case Alice sends an encrypted message M to Bob, she calculates the hash function of the message then, she must convert all the characters of the message $M = \{c_1, c_2, \ldots, c_n\}$ to the points of the elliptic curve by using a code table, which is agreed upon by the two entities Alice and Bob. Then, each point is encrypted on a pair of encryption points C_{a1}, C_{a2} as follows:

- She calculates h = H(M) , where H(M) = SHA-1(M).
- She selects a large random number $k_a <$ ord(G) and computes $C_{a1} = k_a * G$.
- She computes

$C_{a2} = c_i + S_b + k_a * Q_b + n_a * P_a$.

After encrypting all the characters of the message, Alice communicates the hash of the message as well as an encrypted message to Bob in a public channel $\{h, ([C_{a1}, C_{a2}] ; [C_{a1}, C_{a2}] \ldots)\}$.

Algorithm 2 Encryption

Input : p, E, G, Q_i, S_i , n_i ,P_i and M
Output: cipher text (C_{i1}, C_{i2}) and h

If (Alice)
1- Compute h = H(M) // H(.) is hash function
2- Select k_a // $k_a <$ ord (G)
3- Compute $C_{a1} = k_a * G$
4- Compute $C_{a2} = M + S_b + k_a * Q_b + n_a * P_a$
5- Send ((h, C_{a1}, C_{a2}), Bob)

If (Bob)
1- Compute h = H(M) // H(.) is hash function
2- Select k_b // $k_b <$ ord (G)
3- Compute $C_{b1} = k_b * G$
4- Compute $C_{b2} = M + S_a + k_b * Q_a + n_b * P_b$
5- Send ((h , C_{b1}, C_{b2}), Alice)

For decryption, Alice and Bob decrypt their messages following the steps in Algorithm 3. In case Bob receives the hash of the message and an encrypted message, he decrypts it as follows:

- He computes $c_i = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b$.
- He computes H(M) = h' and compares the two hashes if h' is different to h the message has been modified (corrupted) during transmission.

Decryption works out properly

$$c_i = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b$$

$$= c_i + S_b + k_a * Q_b + n_a * P_a - S_a - n_b * C_1 - n_b * P_b$$

$$= c_i + n_b P_b + n_b n_a G + k_a n_b G + n_a P_a - n_a P_a - n_a n_b G - n_b k_a G - n_b P_b$$

$$= c_i$$

Algorithm 3 Decryption

Input : p, E, G, Q_i, S_i , n_i ,P_i and (C_{i1}, C_{i2}, h)
Output: plaintext M or corrupted

If (Alice)
1- Compute $M = C_{b2} – S_b – n_a*C_{b1} – n_a*P_a$
2- if (H(M) = h) Output (M)
else Output 'corrupted'

If (Bob)
1- Compute $M = C_{a2} – S_a – n_b*C_{a1} – n_b*P_b$
3- if (H(M) = h) Output (M)
else Output 'corrupted'

5.1 Security consideration

In this method, to encrypt each character of a message, the sender uses a different random number that allows having different points of the elliptic curve for the same letter. The sender encrypts the message with [23]:

- Receiver's public key in order to ensure the confidentiality of the message,
- Receiver's public key specific for the sender in order to ensure the authenticity of the message because with this specific public key for the sender alone the receiver ensures that the encryption was done by the sender only,
- Sender's private key, in order to digitally sign the message.

Moreover, by communicating the hash of the message, the receiver can check whether the message has been changed or not, which allows ensuring the integrity of the message transmitted. Hence, the cipher has achieved the qualities of confidentiality, integrity, authentication and non-repudiation. Therefore, our cryptosystem is less vulnerable to man-in-middle attack and the security of our cryptosystem is based on the difficulty of the discrete logarithm problem in elliptic curve (ECDLP).

5.2 System test

The following example shows the steps to follow by applying our cryptosystem to message transfer between two communicating parties Alice and Bob.

Let E be an elliptic curve define over Fp.

Where p = 47 and parameters a = 1, b = 4.

The graph of the curve $y^2 = x^3 + x + 4$ is shown below in Fig.2.

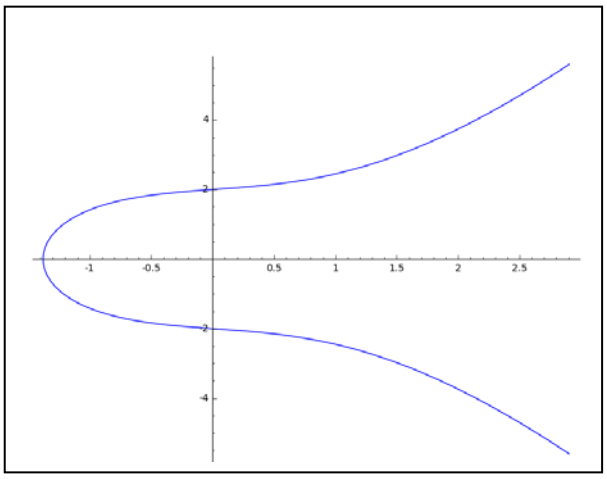

Fig.2.Elliptic Curve $y^2 = x^3 + x + 4$.

The points on the elliptic curve $E_{61}(1, 4)$ are :
{∞, (0 , 2), (0 , 45), (1 , 10), (1 , 37), (2 , 22), (2 , 25), (3 , 9), (3 , 38), (4 , 5), (4 , 42), (7 , 5), (7 , 42), (8 , 17), (8 , 30), (9 , 15), (9 , 32), (10 , 11), (10 , 36), (14 , 6), (14 , 41), (16 , 11), (16 , 36), (20 , 9), (20 , 38), (21 , 11), (21 , 36), (23 , 16), (23 , 31), (24 , 9), (24 , 38), (25 , 12), (25 , 35), (26 , 13), (26 , 34), (27 , 16), (27 , 31), (30 , 3), (30 , 44), (31 , 13), (31 , 34), (34 , 12), (34 , 35), (35 , 12), (35 , 35), (36 , 5), (36 , 42), (37 , 13), (37 , 34), (38 , 21), (38 , 26), (39 , 1), (39 , 46), (41 , 8), (41 , 39), (44 , 16), (44 , 31), (46 , 7), (46 , 40) }.

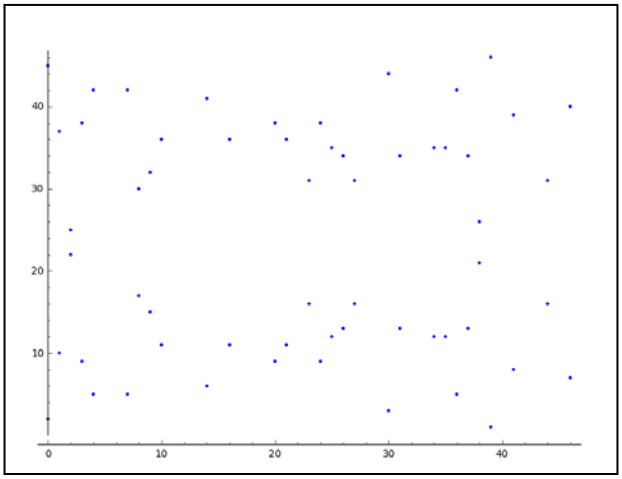

Fig.3. Elliptic Curve $E_{47}(1,4)$.

The points on the elliptic curve over finite field $E_{47}(1,4)$: $(y^2 = x^3 + x + 4) \bmod 47$ is shown above in Fig.3.

Let G=(30,3) is the generator of the cyclic group $E_{61}(1,4)$.

TABLE.1. CORRESPONDING CHARACTERS TO THE CO-ORDINATE POINTS

	a	**b**	**c**	**d**	**e**
∞	(0 , 2)	(0 , 45)	(1 , 10)	(1 , 37)	(2 , 22)
f	**g**	**h**	**i**	**j**	**k**
(2 , 25)	(3 , 9)	(3 , 38)	(4 , 5)	(4 , 42)	(7 , 5)
l	**m**	**n**	**o**	**p**	**q**
(7 , 42)	(8 , 17)	(8 , 30)	(9 , 15)	(9 , 32)	(10 , 11)
r	**s**	**t**	**u**	**v**	**w**
(10 , 36)	(14 , 6)	(14 , 41)	(16 , 11)	(16 , 36)	(20 , 9)
x	**y**	**z**	**0**	**1**	**2**
(20 , 38)	(21 , 11)	(21 , 36)	(23 , 16)	(23 , 31)	(24 , 9)
3	**4**	**5**	**6**	**7**	**8**
(24 , 38)	(25 , 12)	(25 , 35)	(26 , 13)	(26 , 34)	(27 , 16)
9	**,**	**.**	**:**	**(**	**)**
(27 , 31)	(30 , 3)	(30 , 44)	(31 , 13)	(31 , 34)	(34 , 12)
!	**?**	**@**	**&**	**%**	*****
(34 , 35)	(35 , 12)	(35 , 35)	(36 , 5)	(36 , 42)	(37 , 13)
/	**-**	**+**	**^**	**=**	**£**
(37 , 34)	(38 , 21)	(38 , 26)	(39 , 1)	(39 , 46)	(41 , 8)
$	**#**	**;**	**‘**	**ç**	
(41 , 39)	(44 , 16)	(44 , 31)	(46 , 7)	(46 , 40)	

Keys exchange

- Alice selects a random number $n_a = 3$, any point $P_a = (21, 11)$ on the elliptic curve as her secret keys. She computes

  $Q_a = n_a * G = 3(30,3) = (14,41)$ and publishes it as her public key.

- Bob selects a random number $n_b = 7$, any point $P_b = (7, 5)$ on the elliptic curve as his secret key. He computes

  $Q_b = n_b * G = 7(30,3) = (23,31)$ and publishes it as his public key.
- Alice calculates $S_a = n_a*(P_a + Q_b)= 3[(21,11) + (23,31)] = (37,34)$ and publishes it as her specific public key for Bob.
- Bob calculates $S_b = n_b*(P_b + Q_a)= 7[(7,5) + (14,41)] = (10,36)$ and publishes it as his specific public key for Alice.

The keys of Alice and Bob

- Alice's private keys are $n_a = 3$ and $P_a = (21, 11)$.
- Alice's public key is $Q_a = (14,41)$.
- Alice's specific public key for Bob is $S_a = (37,34)$.
- Bob's private keys are $n_b = 7$ and $P_b = (7, 5)$.
- Bob's public key is $Q_b = (23,31)$.
- Bob's specific public key for Alice is $S_b = (10,36)$.

Encryption

If Alice wants to send the message M = "secret" to Bob:

- She calculates the hash function SHA-1(secret) of the message h = 9005453664329993909.
- She must convert all the text characters of the message into points on elliptic curve using agreed upon code table as presented in TABLE.1

secret = {(14,6), (2,22), (1,10), (10,36), (2,22), (14,41)}

- Alice encrypts the first character "s" corresponds to point (14, 26) as follows.
  - She selects a random number $k_a = 12$ and computes
  - $C_{a1} = k_a * G = (0,45)$.
  - $C_{a2} = c_1 + S_b + k_a * Q_b + n_a * P_a = (4,42)$.

  So the character "s" in the plain text is encrypted by the two points [(0,45), (4,42)].
- Alice encrypts the second character "e" corresponds to point (2, 2) as follows.

- She selects a random number k_a= 11 and computes
- $C_{a1} = k_a * G = (44,16)$.
- $C_{a2} = c_2 + S_b + k_a * Q_b + n_a * P_a = (10,11)$.

So the character "e" in the plain text is encrypted by the two points [(44,16), (10,11)].

- Alice encrypts the third character "c" corresponds to point (1, 10) as follows.
  - She selects a random number $k_a = 5$ and computes
  - $C_{a1} = k_a * G = (4,5)$.
  - $C_{a2} = c_3 + S_b + k_a * Q_b + n_a * P_a = (7,5)$.

  So the character "c" in the plain text is encrypted by the two points [(4,5), (7,5)].
- Alice encrypts the fourth character "r" corresponds to point (10,36) as follows.
  - She selects a random number k_a= 17 and computes
  - $C_{a1} = k_a * G = (2, 22)$.
  - $C_{a2} = c_4 + S_b + k_a * Q_b + n_a * P_a = (20,38)$.

  So the character "c" in the plain text is encrypted by the two points [(2,22), (20,38)].
- Alice encrypts the fifth character "e" corresponds to point (2, 22) as follows.
  - She selects a random number k_a= 21 and computes
  - $C_{a1} = k_a * G = (7,42)$.
  - $C_{a2} = c_5 + S_b + k_a * Q_b + n_a * P_a = (9,32)$.

  So the character "c" in the plain text is encrypted by the two points [(7,42), (9,32)].
- Alice encrypts the sixth and last character "t" corresponds to point (14,41) as follows.
  - She selects a random number k_a= 10 and computes
  - $C_{a1} = k_a * G = (24,38)$.
  - $C_{a2} = c_6 + S_b + k_a * Q_b + n_a * P_a = (34,35)$.

  So the character "c" in the plain text is encrypted by the two points [(24,38), (34,35)].

Alice sends the hash of the message as well as an encrypted message {9005453664329993909, [(0,45), (4,42)], [(44,16), (10,11)], [(4,5), (7,5)], [(2,22), (20,38)], [(7,42), (9,32)], [(24,38), (34,35)]} to Bob in public channel.

Decryption

Bob after receiving the cipher text decrypts it as follows.

- $c_1 = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b = (14,6)$ which corresponds to the character "s" in the code table.
- $c_2 = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b = (2,22)$ which corresponds to the character "e" in the code table.
- $c_3 = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b = (1,10)$ which corresponds to the character "c" in the code table.
- $c_4 = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b = (10,36)$ which corresponds to the character "r" in the code table.
- $c_5 = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b = (2,22)$ which corresponds to the character "e" in the code table.
- $c_6 = C_{a2} - S_a - n_b * C_{a1} - n_b * P_b = (14,41)$ which corresponds to the character "t" in the code table. Then "secret" is the plain text.

Bob finds the decrypted message M= 'secret', then he computes the hasher h' of the message M, in order to compare it with the hasher which has received h.

h' = SHA-1(secret)

= 9005453664329993909

= h.

6 Comparison

TABLE 2 shows a comparison between our cryptosystem and the cryptosystem of D. S. Kumar et al. [22] considering the number of operations, the number of points and the security constraints such as confidentiality, integrity, authentication and non-repudiation. P.A denotes the operation of point adding and S.M denotes the operation of scalar multiplication.

The below comparison shows that in our cryptosystem there are fewer scalar multiplication operations than in D. S. Kumar et al [22], which allows us to have a faster cryptosystem and fewer points than D. S. Kumar et al [22], which is beneficial, because it is not always easy to find points on an elliptic curve. Moreover, in our cryptosystem we find that the integrity of the messages is ensured contrary to the cryptosystem of D. S. Kumar et al [22]. Therefore, our cryptosystem is more robust than D. S. Kumar et al [22].

TABLE.2. COMPARISON OF OUR CRYPTOSYSTEM AND D. S. KUMAR ET AL [22].

		Our cryptosystem	**D. S. Kumar et al [22]**
Operations	**P.A**	8	8
	S.M	9	11
Number of points		13	16
Confidentiality		Yes	Yes
Integrity		Yes	No
Authentication		Yes	Yes
Non-repudiation		Yes	Yes

7 Conclusion

In this paper, we proposed a new public-key cryptosystem that ensures all security constraints such as confidentiality, integrity, authenticity, and non-repudiation of the cipher text. In our cryptosystem, the confidentiality of the message is ensured by the public key, the integrity is ensured by the hash function of the message and the authenticity of the message is ensured by the specific public key published by each entity for a specific communicator. Moreover, the private key of the sender guarantees that the message has been implicitly signed by the sender. Therefore, our cryptosystem is less prone to man-in-middle attack and its security is based on the difficulty of the discrete logarithm problem in elliptic curve (ECDLP).

References:

[1] W. Diffie and M. E. Hellman. New directions in cryptography. IEEE Trans. Inform. Theory, IT-22: 644-654, Nov 1976.

[2] A. J. Menezes, T. Okamoto, and S.A. Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field", IEEE Transaction on Information Theory, Vol. 39, No. 5, September 1993.

[3] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transaction on Information Theory, Vol. IT-31, No. 4, July 1985.

[4] N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, Vol. 48, No.177, pp.203-209, Jan 1987.

[5] V.S. Miller, "uses of elliptic curves in cryptography", In Conference on the theory and application of cryptographic techniques. Springer, Berlin, Heidelberg, 1985. p. 417-426.

[6] W.Trappe and L.C. Washington, "Introduction to cryptography with codin theory", 2nd edition. Pearson-Prentice-Hall 2006.

[7] A.S.Haichour and M.Hamadouche, "Elliptic Curve Cryptographic Processor Design using FPGAs", IEEE International Conference on Control, Engineering and Information Technology (CEIT), 2015, p. 1-6. Tlemcen, Algeria.

[8] S. Aifen, L. C.K. Hui, Y. Yixian and K.P. Chow, "Elliptic Curve Cryptography Based Authenticated Key Agreement With PRE-SHARE Password", JOURNAL OF ELECTRONICS(CHINA), Vol.22, No.3, p265-272, May 2005.

[9] A.S.Haichour , M.Hamadouche and A.Khouas, "Hardware Design and Implementation of ElGamal Elliptic Curve Cryptosystem", Wulfenia Journal, Vol 23, No.2, Feb 2016, pages 62-85.

[10] A. M. Johnston, P. S. Gemmell, "Authenticated Key Exchange Provably Secure against the Man-in-the-Middle Attack", Journal of Cryptology, Springer, pages 139-148, 2002.

[11] R.L.Rivest,M.E. Hellman, J.C.Anderson, and J.W. Lyons, "Response to NIST's proposal", Communication of ACM, Vol. 35, No. 7, pp.50–52, 1992.

[12] FIPS-186, the first version of the official DSA specification.

[13] H.Junru, "The improved elliptic curve digital signature algorithm", in International Conference on Electronic & Mechanical Engineering and Information Technology,

pp.257–259, Harbin, China, 12–14 August 2011.

[14] M. K. Chande and C.C Lee, "An improvement of a elliptic curve digital signature algorithm", International Journal of Internet Technology and Secured Transactions, Vol. 6, No.3 pp. 219 – 230, 2016.

[15] W.Diffie, P.C. Van Oorschot and M.J. Wiener "Authentication and Authenticated Key Exchanges", Dignes, Codes and Cryptography, Springer, Vol. 2, No. 2,pages 107-125, June 1992.

[16] Lv. Xixiang, Li. Hui and B. Wang, "Authenticated asymmetric group key agreement based on certificateless cryptosystem", International Journal of Computer Mathematics, Vol. 91, No. 3, May 2013.

[17] K. Al_Sultan, M. Saeb and U. A. Badawi, "A New Two-Pass Key Agreement Protocol", IEEE, pages 509-511, 2014.

[18] K. R Chandrasekhara Pillai1 and M. P Sebastian, "Elliptic Curve based Authenticated Session Key Establishment Protocol for High Security Applications in Constrained Network Environment", International Journal of Network Security & Its Applications, Vol.2, No.3,pages 144-156, July 2010.

[19] L. Rongxing, L. Xiaodong, C. Zhenfu, Q. Liuquan and L Xiaohui, "A simple deniable authentication protocol based on the Diffie–Hellman algorithm", International Journal of Computer Mathematics, Vol. 85, No. 9, pages 1315-1323, September 2008.

[20] K. A. Kumari, G. S. Sadasivam and L. Rohini, "An Efficient 3D Elliptic Curve Diffie–Hellman (ECDH) Based Two-Server Password-Only Authenticated Key Exchange Protocol with Provable Security", IETE Journal of Research, April 2016.

[21] D. S. Kumar, CH. Suneetha and A. Chandrasekhar, "Authentic key transport in symmetric cryptographic protocols using some elliptic curves over finite fields", International Journal of Mathematical Archive, Vol. 3, No.1, pages 137-142, Junary 2012.

[22] D. S. Kumar, CH. Suneetha and A. Chandrasekhar, "Ecryption of data using elliptic curves over finite fields", International Journal of Distributed and Parallel Systems, Vol. 3, No.1, pages 301-308, Junary 2012.

[23] N. Mehibel and M. Hamadouche, "A new algorithm for a public key cryptosystem using elliptic curve", 2017 European Conference on Electrical Engineering and Computer Science, Bern, Switzerland, November 17-19, 2017 to be published.

[24] N. Mehibel and M. Hamadouche. "A new approach of elliptic curve Diffie-Hellman key exchange", 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B), IEEE, 2017. p. 1-6., Boumerdes, Algeria.

[25] National Institute of Standards and Technology. 2012. Secure Hash Standard. NIST FIPS 180–4.