A method for constructing public-key block cipher schemes based on elliptic curves

Luu Hong Dung*

Military Technical Academy.
*Corresponding author: luuhongdung@mta.edu.vn
Received 16 Sep 2022; Revised 2 Dec 2022; Accepted 12 Dec 2022; Published 30 Dec 2022
DOI: https://doi.org/10.54939/1859-1043.j.mst.CSCE6.2022.114-121

ABSTRACT

The article proposes a method for constructing public-key block cipher schemes based on the difficulty of the discrete logarithm problem on elliptic curves. The schemas are construc according to the proposed method and can simultaneously perform security functions and authenticate the origin and integrity of the encrypted message. In addition, a shared secret key is established between the sender/encryptor and the receiver/decryptor for each encrypted message based on public key cryptography which also improves the security of these cipher schemes.

Keywords: Symmetric key cryptography; Public key cryptography; Block cipher; Encryption – Authentication schemes; Discrete logarithm problem on elliptic curves.

1. INTRODUCTION

In [1-3], a solution was proposed for constructing block cipher schemes based on OTP cipher [4]. The benefit of algorithms constructed in accordance with solution is that they inherit the security and efficiency of the OTP cipher [1-3], but the shared secret key between sender/encryptor and receiver/decryptor may be reused several times. Additionally, the construction, management, and distribution of keys are carried out similarly to other symmetric-key cryptosystems currently being applied in practice (DES, AES, etc.). The paper proposes a method for constructing public-key block cipher schemes based on the difficulty of the discrete logarithm problem on elliptic curves. Under this method, a shared secret key is established between the sender/encryptor and the receiver/decrypter for each message to be encrypted based on the mechanism of public key cryptography, which helps improve the security of these cipher schemes. The proposed method here can be applied to block cipher algorithms constructed according to solution in [1-3] as well as to commonly used block cipher algorithms in practice such as: DES, AES, etc.

2. METHOD FOR CONSTRUCTING PUBLIC-KEY BLOCK CIPHER SCHEMES

2.1. Proposed method

The method for constructing public key block cipher schemes proposed here includes the Key Generation Algorithm, the Encryption Algorithm and the Decryption – Authentication Algorithm, described as follows:

2.1.1. Key Generation Algorithm

The End User's key is generated by the key generation algorithm from the set of domain parameters, which includes:

- **p** is a prime number specifying the underlying finite field F_p.
- **E**(F_p) is Elliptic curve defined on the finite field F_p by equation E(a,b):

$$y^2 = x^3 + ax + b \text{ with: } a, b \in F_p \text{ and satisfied: } 4a^3 + 27b^2 \neq 0 \bmod q$$

- **G** is the base point in $E(F_p)$.

- **q** is the order of G in $E(F_p)$.

Attention:

In order for the discrete logarithm problem to be difficult to solve on $E(F_p)$, the domain parameter set can be selected according to ISO/IEC 15946 [5], ANSI X9.62 [6] or FIPS PUB 186-4 [7].

The **p**, **a**, **b**, **G**, **q** parameters are system parameters or domain parameters generated by the service provider and (**d**,**P**) are the secret, public key pair of the End User (sender/encryptor, receiver/decryptor). The Key generating algorithm is described as follows:

Algorithm 1.1:
input: $E(F_p) = (p, a, b, G, q)$.
output: (d,P).

[1]. Generate the secret key d in the range (1,q):

$d = RNG(\{1,2,\ldots,q-1\})$

[2]. Calculate the public key P by:

$P = (x_p, y_p) = d \cdot G$

Notes:

- RNG(): Random or pseudo-random number generator.

- (x_p, y_p): The coordinates of the point P on $E(F_p)$.

Suppose, d_s is the secret key of the sender (encryptor) and d_r is the secret key of the receiver (decryptor), then the corresponding public keys of the sender are:

$$P_s = (x_{ps}, y_{ps}) = d_s.G$$

and of the receiver are:

$$P_r = (x_{pr}, y_{pr}) = d_r.G$$

2.1.2. Encryption algorithm

Algorithm 1.2:
input: $E(F_p) = (p, a, b, G, q)$, d_s, P_r, M_1.
output: (R,C).

[1]. Calculate the S_e according to the formula:

$S_e = (x_{se}, y_{se}) = d_s \cdot P_r$

[2]. Calculate the value R follow:

$R = F_1(M_1, x_{se})$

[3]. Calculate the sender's encryption key K_e:

$K_e = F_1(R, x_{se})$

[4]. Encrypt the message to be sent M_1 according to:

$C = Encrypt(K_e, M_1)$

[5]. Send ciphertext (R,C) to the receiver.

Notes:

- $F_1()$: Cryptographic hash function, eg: SHA-1/SHA256 [8], etc.

- (x_{se}, y_{se}): Coordinates of the point S_e on $E(F_p)$.

In this scheme, *Encrypt()* is an encryption function with a symmetric key K_e

constructed according to the solution in [1-3] then the plaintext M is encrypted as *n* data blocks M_i of size *m* bits:

$$M = \{M_1, M_2,\ldots, M_n\}$$

The output of *Encrypt()* which is the C component of the ciphertext also includes *n* data blocks C_i of size m bits:

$$C = \{C_1, C_2,\ldots, C_n\}$$

One time use key K_{OT} consists of *n* subkeys K_i whose size corresponds to the size of the plaintext block:

$$K_{OT} = \{K_1, K_2,\ldots, K_n\} \text{ với: } K_1 = K_e$$

The encryption function *Encrypt()* is described as follows:

Algorithm 1.3:
input: $M = \{M_1, M_2,\ldots, M_n\}$, K_e.
output: $C = \{C_1, C_2,\ldots, C_n\}$.

[1]. $K_1 = K_e$
[2]. for i = 1 to n do
begin
$C_i = M_i \oplus K_i$
$K_{i+1} = F_2(M_i, K_i)$
end
[3]. return C

Notes:

- The operation $\oplus$ is the addition modulo 2 (XOR) of two bit strings.
- $F_2()$ is a Random or Pseudo-random number generator function.

2.1.3. Decryption – Authentication Algorithm

Algorithm 1.4:
input: $E(F_p) = (p, a, b, G, q)$, d_r, P_s, (R,C).
output: M_2.

[1]. Calculate the S_d according to the formula:
$S_d = (x_{sd}, y_{sd}) = d_r . P_s$
[2]. Calculate receiver's decryption key K_d:
$K_d = F_1(R, x_{sd})$
[3]. Decrypt the received message C according to:
$M_2 = \text{Decrypt}(K_d, C)$
[4]. Calculate the value V according to:
$V = F_1(M_2, x_{sd})$
[5]. Checks if: V = R then: $M_2 = M_1$, means that the origin and integrity of the post-decrypted message is confirmed.

Note:

- (x_{sd}, y_{sd}): Coordinates of the point S_d on $E(F_p)$.

The decryption function with the symmetric key *Decrypt()* is constructed according to the solution in [1-3] with the input as the C component of the ciphertext and the shared secret key K, the output is the post-decrypted message M consisting of *n* data block of size *m* bits:

$$M = \{M_1, M_2,\ldots, M_n\}$$

One time use key K_{OT} is similar to the sender/encryption side, consisting of *n* subkeys of the size of the plaintext block:

$$K_{OT} = \{K_1, K_2,\ldots, K_n\} \text{ with: } K_1 = K_d$$

The decryption function *Decrypt()* then has the form:

Algorithm 1.5:
input: $C = \{C_1, C_2,\ldots, C_n\}$, K_d.
output: $M = \{M_1, M_2,\ldots, M_n\}$.
[1]. $K_1 = K_d$
[2]. for i = 1 to n do
begin
$M_i = C_i \oplus K_i$
$K_{i+1} = F_2(M_i, K_i)$
end
[3]. return M

2.1.4. The correctness of the proposed scheme

What needs to be proved here is: if the received ciphertext is the same as the sent ciphertext, then the message after decryption is also the message before encryption: $M_2 = M_1$ and the condition: V = R will be satisfied. Therefore, after decryption if the condition: V = R is satisfied, the receiver can confirm with certainty the origin and integrity of the received message.

We have:

$$S_d = d_r . P_s = d_r . (d_s . G) = d_s . (d_r . G) = d_s . P_r = S_e$$

Deduce: $x_{sd} = x_{se}$

So we also have:

$$K_d = F_1(R, x_{sd}) = F_1(R, x_{se}) = K_e$$

Therefore, we have the first proof:

$$M_2 = \text{Decrypt}(K_d, C) = \text{Decrypt}(K_d, \text{Encrypt}(K_e, M_1))$$
$$= \text{Decrypt}(K_d, \text{Encrypt}(K_d, M_1)) = M_1$$

Then, we have the second proof:

$$V = F_1(M_2, x_{sd}) = F_1(M_1, x_{se}) = R$$

2.2. An application scheme

An application implementation of the proposed method is to use the SHA-1 hash function [8] to perform the roles of functions F_1 and F_2. In this scheme, the plaintext M_1 is encrypted as *n* data blocks of size 160 bits:

$$M_1 = \{M_{11}, M_{12}, \ldots, M_{1i}, \ldots, M_{1n}\}, i = \overline{1,n}, |M_{1i}| = 160 \text{ bits}$$

The sent ciphertext consists of two components R and C. Where, the size of R corresponds to the size of the SHA-1 output data (160 bits) and C consists of *n* blocks of data, each of 160 bits in size:

$$C = \{C_1, C_2,\ldots, C_i,\ldots, C_n\}, i = \overline{1,n}, |C_i| = 160 \text{ bits}$$

Key K_{OT} consists of *n* subkeys K_i also 160 bits in size with $K_1 = K_e$:

$$K_{OT} = \{K_1, K_2, \ldots, K_i, \ldots, K_n\}, i = \overline{1,n}, |K_i| = 160 \text{ bits}$$

The decrypted message M_2 can be received as *n* blocks of data, each of 160 bits in size:

$$M_2 = \{M_{21}, M_{22}, \ldots, M_{2i}, \ldots, M_{2n}\}, i = \overline{1,n}, |M_{2i}| = 160 \text{ bits}$$

Then the encryption and decryption algorithms of the scheme can be described in detail as follows:

Algorithm 2.1 : Encryption.
input: $E(F_p) = (p, a, b, G, q)$, d_s, P_r, M_1.
output: (R, C).

[1]. Calculate the S_e according to the formula:
$S_e = (x_{se}, y_{se}) = d_s . P_r$
[2]. Calculate the value R follow:
$R = \text{SHA-1}(M_1 || x_{se})$
[3]. Calculate the sender's encryption key K_e:
$K_e = \text{SHA-1}(R || x_{se})$
[4]. $K_1 = K_e$
for i = 1 to n do
begin
$C_i = M_{1i} \oplus K_i$
$K_{i+1} = \text{SHA-1}(M_{1i} || K_i)$
end
[5]. Send ciphertext (R,C) to the receiver.

Note:

- The operation "||" is the concatenation operator of two bit strings.

Algorithm 2.2 : Decryption – Authentication.
input: $E(F_p) = (p, a, b, G, q)$, d_r, P_s, (R,C).
output: M_2.

[1]. Calculate the S_d according to the formula:
$S_d = (x_{sd}, y_{sd}) = d_r . P_s$
[2]. Calculate the value of the decryption key K_d:
$K_d = \text{SHA-1}(R || x_{sd})$
[3]. $K_1 = K_d$
for i = 1 to n do
begin
$M_{2i} = C_i \oplus K_i$
$K_{i+1} = \text{SHA-1}(M_{2i} || K_i)$
end
[4]. Calculate the value of V according to:
$V = \text{SHA-1}(M || S_d)$
[5]. Check if: V = R then return the result: $M_2 = \{M_{21}, M_{22}, \ldots, M_{2n}\}$.
Otherwise, if: $V \neq R$ then: return $M_2 = \{0,0,\ldots,0\}$.

Note:

- When receiving the message: $M_2 = \{0,0,\ldots,0\}$ after decryption, the receiver assumes that the message is tampered or a communication error has occurred. Otherwise, this is the encrypted message.

2.3. Some evaluation of the security level of the proposed scheme

The security level of the proposed scheme is assessed by its ability to resist some typical attacks as follows:

Ciphertext-only attack: To decrypt a message, an attacker needs to compute either the encryption key e or the decryption key d. First, the attacker needs to find out the sender 's secret key d_s to calculate S_e by:

$$S_e = (x_{se}, y_{se}) = d_s . P_r$$

or find out the secret key d_r of the receiver to calculate S_d:

$$S_d = (x_{sd}, y_{sd}) = d_r . P_s$$

Then calculate the encryption key:

$$K_e = F_1(R, x_{se})$$

or calculate the decryption key:

$$K_d = F_1(R, x_{sd})$$

However to calculate d_s from:

$$P_s = d_s . G$$

or d_r from:

$$P_r = d_r . G$$

The attacker needs to solve the discrete logarithm problem on $E(F_p)$. Currently, no polynomial–time algorithm has been published for this difficult problem [9].

- *Known-plaintext attack:* In this case, it makes no sense to calculate e or d, because this key is used only once for an encrypted message. But the attacker can still find S_e or S_d to calculate e or d for later encryption sessions. Then, in addition to solving the discrete logarithm problem on $E(F_p)$ like the above case, the attacker can also rely on the public message m_1 to calculate S_e according to:

$$R = F_1(M_1, x_{se})$$

However, in this way, the attacker cannot achieve his goal because of the one-way nature of the hash function.

- *Spoofing attack*: In the proposed scheme, an attacker who wants to impersonate a certain sender to send a forged message to the receiver needs to obtain the secret parameter S_e or S_d of the sender or receiver. But from the above analysis, it is not possible if the attacker cannot solve the discrete logarithm problem on $E(F_p)$ or the problem of the one-wayness of the hash function. Furthermore, the post–decrypted message is only authenticated for its origin and integrity when the following conditions are satisfied:

$$F_1 (M_2, x_{sd}) = F_1(M_1, x_{se})$$

Due to the collision resistance of the hash function, to satisfy the above condition, it is

necessary to satisfy the following two conditions simultaneously: $M_2 = M_1$ and: $S_d = S_e$. With the first condition: $M_2 = M_1$ the receiver can fully confirm the integrity of the message after decryption, and the origin of the message is authenticated based on the condition: $S_d = S_e$ as follows: Since the receiver uses the public key P_s of the sender to generate S_d follow:

$$S_d = d_r . P_s$$

Should be to: $S_d = S_e$ then S_e must be generated from the sender's secret key d_s by:

$$S_e = d_s . P_r$$

Only the owner of the public key y_s knows the corresponding secret key d_s, i.e. only the owner of the public key P_s is capable of generating S_e equal to S_d of the receiver, which allows the receiver to verify that the source of the decrypted message was generated by the owner of the public key P_s. When an attacker sends a spoofed message to a receiver using a value different from the key d_s of the sender it is impersonating (because the attacker does not know the d_s of the impersonated sender), the value S_d generated by the receiver will be different from the S_e of the impostor, resulting in the message being rejected.

3. CONCLUSIONS

The article proposes a method for constructing block cipher schemes based on the mechanism of public key cryptography. The advantage of encryption schemes based on this method is that although the security and efficiency of the OTP are preserved, but the shared secret key is only used to encrypt each message. These are very important properties for these cipher schemes to be applicable in practice. Additionally, because of the process for authenticating the origin and integrity of the encrypted message, these cipher schemes are resistant to spoofing attacks, which is one of the fundamental requirements for real-world applications.

REFERENCES

[1]. Luu Hong Dung, Nguyen Anh Viet. *"A solution to build a symmetric-key cryptosystem"*. Information Security Magazine, Issue 5 (057) (2020).

[2]. Luu Hong Dung, Tong Minh Duc, Bui The Truyen. *"Variant of OTP cipher with symmetric-key solution"*. Journal of Science and Technique - Section on Information and Communication Technology (ICT) - No. 16 (2020), Le Quy Don Technical University. ISSN: 1859 - 0209. DOI: 10.56651/lqdtu.jst.v9.n02.210.ict

[3]. Luu Hong Dung, Nguyen Anh Viet, Doan Thi Bich Ngoc. *An encryption and authentication algorithm developed based on the one – time pad cipher*. Journal of Military Science and Technology, ISSN: 1859 - 1403. (2020). DOI: 10.54939/1859-1043.j.mst.87-93.

[4]. Gilbert Vernam . *US Patent 1,310,719*. (1919).

[5]. ISO/IEC 15946: Information technology – Security techniques – Cryptographic Techniques Based on Elliptic Curves, (1999).

[6]. ANSI X9.62. Public Key Cryptography for the Financial Services Industry: Elliptic Cuve Digital Signature Algorithm (ECDSA), (1999).

[7]. National Institute of Standards and Technology, NIST FIPS PUB 186-4. Digital Signature Standard, U.S. Department of Commerce, (2013).

[8]. National Institute of Standards and Technology, NIST FIPS PUB 180-1. (1995).

[9]. Lawrence C. Washington. *"Elliptic curves – Number Theory and Cryptography"*. Chapman & Hall/CRC, (2003).

TÓM TẮT

Một phương pháp xây dựng các lược đồ mã khối khóa công khai dựa trên đường cong elliptic

Bài báo đề xuất phương pháp xây dựng các lược đồ mã khối khóa công khai dựa trên độ khó của bài toán logarit rời rạc trên đường cong elliptic. Các lược đồ được xây dựng theo phương pháp đề xuất ở đây có thể đồng thời thực hiện chức năng bảo mật và xác thực nguồn gốc và tính toàn vẹn của thông điệp mã hóa. Ngoài ra, một khóa bí mật dùng chung được thiết lập giữa người gửi/người mã hóa và người nhận/người giải mã cho mỗi tin nhắn được mã hóa dựa trên mật mã khóa công khai, điều này cũng giúp cải thiện tính bảo mật của lược đồ mã khối.

Từ khoá: Mật mã khóa đối xứng; Mật mã khóa công khai; Mã khối; Các lược đồ mã hóa – xác thực; Bài toán logarit trên đường cong elliptic.