A method for constructing public - key block cipher schemes based on discrete logarithm problem

Luu Hong Dung[1*], Nguyen Vinh Thai[2], Nguyen Kim Thanh[1], Pham Van Hiep[3]

[1]Military Technical Academy;
[2]Academy of Military Science and Technology;
[3]Ha Noi University of Industry.
[*]Corresponding author: luuhongdung@mta.edu.vn
Received 06 Oct. 2023; Revised 07 Dec. 2023; Accepted 12 Dec. 2023; Published 30 Dec. 2023.
DOI: https://doi.org/10.54939/1859-1043.j.mst.CSCE7.2023.15-26

ABSTRACT

The paper proposes a method for constructing block cipher schemes that enable verifying the source and integrity of the encrypted message. Additionally, the shared secret key between the sender/encryptor and the receiver/decryptor for each encrypted message is established based on the mechanism of public key cryptography.

Keywords: Symmetric key cryptography; Public key cryptography; Encryption – authentication scheme; Discrete logarithm problem; OTP cipher.

1. INTRODUCTION

In [1], [2] and [3], a solution was proposed for constructing a symmetric-key cryptosystem based on OTP or Vernam cipher [4]. This paper proposes a method for constructing block cipher schemes based on the solution in [1], [2] and [3] can simultaneously perform security functions and authenticate the origin and integrity of the encrypted message. On the other hand, a shared secret key between the sender/encryptor and the receiver/decryptor for each encrypted message is established based on the mechanism of public key cryptography, which helps improve the security of these cipher schemes. In addition, the method proposed here is applicable to all other symmetric-key cryptographic algorithms, including block and stream ciphers.

2. THE METHOD FOR CONSTRUCTING PUBLIC-KEY BLOCK CIPHER SCHEMES

2.1. Proposed method

The proposed method here includes Parameter and Key Generation algorithm (Algorithm 1.1), Encryption algorithm (Algorithm 1.2 & 1.3) and Decryption – Authentication algorithm (Algorithm 1.4 & 1.5). These algorithms are described as follows:

2.1.1. Key generation algorithm

The end-user's private/public key is established based on a set of domain/system parameters, which include:

- The prime numbers **p** and **q**, satisfy: q|(p - 1). Here, to ensure the security of the scheme, **p** and **q** can be generated as specified in FIPS 186 – 4 [5] or GOST R34.10 – 94 [6].

- An element **g** of prime order **q** is generated according to the formula:

$$g = \alpha^{\frac{p-1}{q}} \bmod p \text{, here: } \boldsymbol{\alpha} \text{ is a value in the range (1,p)}$$

The private key is a randomly selected value in the range (1,q), the corresponding public key is generated from the private key and domain/system parameters by the Key generation algorithm (**Algorithm 1.1**) as follows:

Algorithm 1.1:

input: p, q, g.

output: x, y.

[1]. Select a secret key *x* in the range (1,q).

[2]. Calculate the public key y calculate g according to the formula:

$$y = g^x \bmod p$$

Note:

– p, q, g: The system parameters.

– x, y: The private key, the public key of the end-user in the system.

Assume that x_s is the private key of the sender/encryptor and x_r is the private key of the receiver/decryptor, then the corresponding public key of the sender is:

$$y_s = g^{x_s} \bmod p$$

and the public key of the receiver is:

$$y_r = g^{x_r} \bmod p$$

2.1.2. Encryption algorithm

In the schemes constructed according to the method proposed here, the Encryption algorithm takes as input the plaintext P, the sender's private key x_s, the receiver's public key y_r and system parameters. The output of the algorithm is the ciphertext C and the component R. Here, the component R is used to generate a secret key shared between the sender and the receiver.

The Encryption algorithm (**Algorithm 1.2**) is described in pseudocode as follows:

Algorithm 1.2:

input: g, p, q, x_s, y_r, P.

output: (R,C).

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

$$S_e = \left(y_r\right)^{x_s} \bmod p$$

[2]. Calculate the value of R by the function $F_1()$:

$R = F_1(P,S_e)$

[3]. Calculate the value of the encryption key K_e (shared secret key of the sender) by the function $F_2()$:

$K_e = F_2(R,S_e)$

[4]. Use the encryption function with the symmetric key $E_K()$ to encrypt the plaintext P:

$C = E_{Ke}(P)$

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

Note:

– P: The plaintext.

– (R,C): The ciphertext corresponding to P.

Attention:

– $F_1()$: The function should be constructed so that calculating S_e is difficult even when P is known.

– $F_2()$: The function should be constructed so that calculating S_e is difficult even when K_e is known.

In this scheme, $E_K()$ is an encryption function with a symmetric key K_e constructed according to the solution in [1], [2] and [3], then the plaintext P is encrypted as *n* data blocks P_i of size *m* bits:

$$P = \{P_1, P_2,\ldots, P_n\}$$

The output of $E_K()$ 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 :

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

Note:

In the schemes constructed according to the proposed method, the Encryption function $E_K()$ can also be the encryption function of any block cipher algorithm such as DES, AES.

The Symmetric-Key Encryption function $E_K()$ is described as follows:

Algorithm 1.3:

input: $P = \{P_1, P_2,\ldots, P_n\}$, K.

output: $C = \{C_1, C_2,\ldots, C_n\}$.

[1]. $K_1 = K$

[2]. for i = 1 to n do

begin

$C_i = P_i \oplus K_i$

$K_{i+1} = F_3(P_i, K_i)$

end

[3]. return C

Note:

– The operation $\oplus$ is the addition modulo 2 (XOR) of two-bit strings.

– $F_3()$ is a Pseudo-random or Random number generator function.

2.1.3. Decryption – Authentication algorithm

The Decryption – Authentication algorithm takes as input the ciphertext C, the receiver's private key x_r, the sender's public key y_s and system parameters. The output of the algorithm is the plaintext M. Furthermore, in the schemes constructed according to the proposed method, M is also authenticated in terms of both the origin and data integrity of the message.

The Decryption – Authentication algorithm (**Algorithm 1.4**) is described in pseudocode as follows:

Algorithm 1.4:

input: p, q, g, x_r, y_s, (R,C).

output: M.

[1]. Calculate the value of S_d according to:

$S_d = (y_s)^{xr} \bmod p$

[2]. Calculate the value of decryption key K_d (shared secret key of the receiver) by the function $F_2()$:

$K_d = F_2(R, S_d)$

[3]. Use the decryption function with the symmetric key $D_K()$ to decrypt C:

$M = D_{Kd}(C)$

[4]. Calculate the value of V by the function $F_1()$:

$V = F_1(M, S_d)$

[5]. Check if: V = R then: M = P, means that the origin and integrity of the post-decrypted message are confirmed.

Attention:

– $F_1()$: The function should be constructed so that calculating S_d is difficult even when M is known.

– $F_2()$: The function should be constructed so that calculating S_d is difficult even when K_d is known.

The decryption function with the symmetric key $D_K()$ is constructed according to the solution in [1], [2] and [3] with the input as the C component of the ciphertext and the shared secret key K_d, 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:

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

Note:

In the schemes constructed according to the proposed method, the Decryption function $D_K()$ can also be the encryption function of any block cipher algorithm such as DES, AES.

The Symmetric-Key Decryption function $D_K()$ then has the form:

Algorithm 1.5:

input: C = {C_1, C_2,…, C_n}, K.

output: M = {M_1, M_2,…, M_n}.

[1]. $K_1 = K$

[2]. for i = 1 to n do

begin

$M_i = C_i \oplus K_i$

$K_{i+1} = F_3(M_i,K_i)$

end

[3]. return M

2.1.4. The correctness of the proposed method

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

Indeed, we have:

$$S_d = \left(y_s\right)^{x_r} \bmod p = \left(g^{x_s} \bmod p\right)^{x_r} \bmod p$$

$$= \left(g^{x_r} \bmod p\right)^{x_s} \bmod p = \left(y_r\right)^{x_s} \bmod p = S_e$$

So:

$$K_d = F_2(R,S_d) = F_2(R,S_e) = K_e$$

Therefore, we have the first proof:

$$M = D_{Kd}(C) = D_{Kd}(E_{Ke}(P)) = D_{Ke}(E_{Ke}(P)) = P$$

Then, we have the second proof:

$$V = F_1(M,S_d) = F_1(P,S_e) = R$$

2.2. Some application schemes

2.2.1. The first scheme

An application implementation of the proposed new method is to use the MD5 hash function as mentioned in [2] to perform the role of functions $F_1()$, $F_2()$ and $F_3()$. In this scheme, the plaintext P is encrypted as *n* data blocks of size 128 bits:

$$P = \{P_1,P_2,\ldots,P_i,\ldots,P_n\}, \forall i = \overline{1,n}, |P_i| = 128 \text{ bit}$$

The sent ciphertext consists of two components R and C. Where R has the size of the output data of MD5 (128 bits), and C consists of *n* blocks of data, each of 128 bits in size:

$$C = \{C_1,C_2,\ldots,C_i,\ldots,C_n\}, \forall i = \overline{1,n}, |C_i| = 128 \text{ bit}$$

Key K_{OT} consists of *n* subkeys K_i also 128 bits in size with $K_1 = K_e$:

$$K_{OT} = \{K_1,K_2,\ldots,K_i,\ldots,K_n\}, \forall i = \overline{1,n}, |K_i| = 128 \text{ bit}$$

The decrypted message M can be received as *n* blocks of data, each of 128 bits in size:

$$M = \{M_1, M_2, \ldots, M_i, \ldots, M_n\}, \forall i = \overline{1,n}, |M_i| = 128 \text{ bit.}$$

Then, the Encryption algorithm (**Algorithm 2.1**) of the schema can be described in detail as follows:

Algorithm 2.1:

input: g, p, q, x_s, y_r, P.

output: (R,C).

[1]. Calculate the value of S_e according to:

$$S_e = (y_r)^{x_s} \bmod p$$

[2]. Calculate the value of K_s by:

$K_s = MD5(P||S_e)$

[3]. Calculate the value of R according to:

$$R = g^{K_s} \bmod p$$

[4]. Calculate the value of K_{se} according to:

$$K_{se} = (y_r)^{K_s} \bmod p$$

[5]. Calculate the value of the encryption key K_e by:

$K_e = MD5(R||K_{se})$

[6]. $K_1 = K_e$

for i = 1 to n do

begin

$C_i = P_i \oplus K_i$

$K_{i+1} = MD5(P_i||K_i)$

end

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

and the Decryption– Authentication algorithm (**Algorithm 2.2**) of the schema is described as follows:

Algorithm 2.2:

input: p, q, g, x_r, y_s, (R,C).

output: M.

[1]. Calculate the value of S_d according to:

$$S_d = (y_s)^{x_r} \bmod p$$

[2]. Calculate the value of K_{sd} according to:

$$K_{sd} = R^{x_r} \bmod p$$

[3]. Calculate the value of the decryption key K_d by:

$K_d = MD5(R||K_{sd})$

[4]. $K_1 = K_d$

for i = 1 to n do

begin

$M_i = C_i \oplus K_i$

$K_{i+1} = MD5(M_i||K_i)$

end

[5]. Calculate the value of K_r by:

$K_r = MD5(M||S_d)$

[6]. Calculate the value of V according to:

$V = g^{K_r} \bmod p$

[7]. Check if: V = R then return the result: M = {M_1, M_2,…, M_n}. Otherwise, if: V ≠ R then: return M = {0, 0,…, 0}.

Note:

– When receiving the message: M = {0,0,…,0} after decryption, the receiver assumes that the message is tampered or a communication error has occurred. Otherwise, this is the encrypted message.

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

The correctness of the proposed scheme is proved as follows:

Indeed, we have:

$$K_{sd} = R^{x_r} \bmod p = \left(g^{K_s} \bmod p\right)^{x_r} \bmod p$$

$$= \left(g^{x_r} \bmod p\right)^{K_s} \bmod p = \left(y_r\right)^{K_s} \bmod p = K_{se}$$

So:

$$K_d = MD5(R\ ||K_{sd}) = MD5(R||K_{se}) = K_e$$

Therefore, we have the first proof:

$$M = D_{Kd}(C) = D_{Kd}(E_{Ke}(P)) = D_{Ke}(E_{Ke}(P)) = P$$

We also have:

$$S_d = \left(y_s\right)^{x_r} \bmod p = \left(g^{x_s} \bmod p\right)^{x_r} \bmod p$$

$$= \left(g^{x_r} \bmod p\right)^{x_s} \bmod p = \left(y_r\right)^{x_s} \bmod p = S_e$$

So:

$$K_r = MD5(M||S_d) = MD5(P||S_e) = K_s$$

Then, we have the second proof:

$$V = g^{K_r} \bmod p = g^{K_s} \bmod p = R$$

2.2.2. The second scheme

Another application implementation of the proposed new method is to use the SHA-256 hash function as mentioned in [3] to perform the role of functions $F_1()$, $F_2()$ and $F_3()$. In this scheme, the plaintext P is encrypted as *n* data blocks of size 256 bits:

$$P = \{P_1, P_2, \ldots, P_i, \ldots, P_n\},\ \forall i = \overline{1,n},\ |P_i| = 256 \text{ bit}$$

The sent ciphertext consists of two components R and C. Where R has the size of the output data of SHA-256 (256 bits), and C consists of *n* blocks of data, each of 256 bits in size:

$$C = \{C_1, C_2, \ldots, C_i, \ldots, C_n\},\ \forall i = \overline{1,n},\ |C_i| = 256 \text{ bit}$$

Key K_{OT} consists of *n* subkeys K_i also 256 bits in size with $K_1 = K_e$:

$$K_{OT} = \{K_1, K_2, \ldots, K_i, \ldots, K_n\},\ \forall i = \overline{1,n},\ |K_i| = 256 \text{ bit}$$

The decrypted message M can be received as *n* blocks of data, each of 256 bits in size:

$$M = \{M_1, M_2, \ldots, M_i, \ldots, M_n\},\ \forall i = \overline{1,n},\ |M_i| = 256 \text{ bit.}$$

Then the Encryption algorithm (**Algorithm 3.1**) of the schema can be described in detail as follows:

Algorithm 3.1:

input: g, p, q, x_s, y_r, P.

output: (R,C).

[1]. Calculate the value of S_e according to:

$$S_e = \left(y_r\right)^{x_s} \bmod p$$

[2]. Calculate the value of K_s by:

K_s = SHA-256(P||S_e)

[3]. Calculate the value of R according to:

$$R = g^{K_s} \bmod p$$

[4]. Calculate the value of K_{se} according to:

$$K_{se} = R^{S_e} \bmod p$$

[5]. Calculate the value of the encryption key K_e by:

K_e = SHA-256(K_{se})

[6]. $K_1 = K_e$

for i = 1 to n do

begin

$C_i = P_i \oplus K_i$

K_{i+1} = SHA-256(P_i||K_i)

end

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

and the Decryption – Authentication algorithm (**Algorithm 3.2**) of the schema can be described as follows:

Algorithm 3.2:

input: p, q, g, x_r, y_s, (R, C).

output: M.

[1]. Calculate the value of S_d:

$$S_d = \left(y_s\right)^{x_r} \bmod p$$

[2]. Calculate the value of K_{sd} according to:

$$K_{sd} = R^{S_d} \bmod p$$

[3]. Calculate the value of the decryption key K_d by:

K_d = SHA-256(K_{sd})

[4]. $K_1 = K_d$

for i = 1 to n do

begin

$M_i = C_i \oplus K_i$

K_{i+1} = SHA-256($M_i||K_i$)

end

[5]. Calculate the value of K_r by:

K_r = SHA-256($M||S_d$)

[6]. Calculate the value of V according to:

$$V = g^{K_r} \bmod p$$

[7]. Check if: V = R then return the result: M = {M_1,M_2,…,M_n}. Otherwise, if: V ≠ R then: return M = {0,0,…,0}.

The correctness of the proposed scheme is proved as follows:

Indeed, we have:

$$S_d = \left(y_s\right)^{x_r} \bmod p = \left(g^{x_s} \bmod p\right)^{x_r} \bmod p$$

$$= \left(g^{x_r} \bmod p\right)^{x_s} \bmod p = \left(y_r\right)^{x_s} \bmod p = S_e$$

So:

$$K_{sd} = R^{S_d} \bmod p = R^{S_e} \bmod p = K_{se}$$

Infer that:

K_d = SHA-256(K_{sd}) = SHA-256(K_{se}) = K_e

Therefore, we have the first proof:

$$M = D_{Kd}(C) = D_{Kd}(E_{Ke}(P)) = D_{Ke}(E_{Ke}(P)) = P$$

We also have:

$$K_r = \text{SHA-256}(M||S_d) = \text{SHA-256}(P||S_e) = K_s$$

Then, we have the second proof:

$$V = g^{K_r} \bmod p = g^{K_s} \bmod p = R$$

2.3. Some evaluations of the security of the schemas are constructed according to the proposed method

The security of the proposed schemes is evaluated based on their ability to resist some typical attacks as follows:

- *Secret Key Attack:* To be able to compute the secret key of the end-user in the system, the attacker needs to solve the discrete logarithm problem on the finite field F_p [7-20]. Currently, no polynomial time algorithm has been published for this difficult problem.

- *Ciphertext-only Attack*: The analysis in [1], [2] and [3] has shown that direct attack on the functtions $E_K()$, $D_K()$ is not viable when only ciphertext is available. Attacking the functions $F_1()$ and $F_2()$ which are constructed as in sections 2.2.1 and 2.2.2, the attacker is also forced to solve the discrete logarithm problem on the finite field.

- *Known-plaintext attack*: Calculating K_e or K_d makes no sense in this scenario, because these keys are only utilized once. However, the attacker can still find S_e or S_d to calculate K_e or K_d for later encryption sessions. The attacker can rely on known R, P, K_e and K_d to calculate S_e and S_d from:

$$R = F_1(P,S_e)$$

or:

$$K_e = F_2(R,S_e)$$

or:

$$K_d = F_2(R,S_d)$$

However, if the functions $F_1()$ and $F_2()$ are constructed as shown in the application schemas in sections 2.2.1 and 2.2.2, the attacker will not be able to achieve his goal.

Spoofing attack: The OTP cipher does not provide verification for an encrypted message, so an attacker could block the ciphertext was sent and send the recipient a fake ciphertext of the same size as the true message. In the case of decrypting to a meaningless plaintext, the receiver may speculate that the tampering was made or caused by a communication error. However, if decrypted to a meaningful plaintext, then the receiver has no way of confirming whether the plaintext is true or fake. In the schemes are constructed according to the proposed method, the origin and integrity of the message after decryption will be verified if the condition: V = R is satisfied. From calculating the values of M, V and R, the above condition is satisfied only when the following conditions are simultaneously satisfied: $S_d = S_e$ and M = P. Obviously, the condition: $S_d = S_e$ allows the sender and the receiver of the message to verify each other's identity. The condition: M = P allows the receiver to verify the integrity of the post-decrypted message. That allows these schemes to be resistant to spoofing attacks.

3. CONCLUSIONS

The paper proposes a method for constructing block cipher schemes based on OTP cipher and public key cryptography. The advantage of the encryption schemes constructed by this method is that it inherits the security and performance of the OTP cipher. On the other hand, the shared secret key between the sender/encryptor and the receiver/decryptor for each message to be encrypted is established based on the mechanism of public key cryptography. In addition, by validating the origin and integrity of the encrypted message, these encryption schemes are resistant to spoofing attacks. These are very important properties for these block cipher schemes to be applicable in practice. The proposed method here, in addition to being applied to the block cipher algorithms constructed according to the solution in [1], [2] and [3] as presented, is also applicable to all other symmetric-key cryptographic algorithms.

REFERENCES

[1]. Lưu Hồng Dũng, Nguyễn Ánh Việt. *"Một giải pháp xây dựng hệ mật khóa đối xứng"*. Tạp chí An toàn Thông tin, ISSN 1859 - 1256, Số 5 (057), (2020) (in Vietnamese).

[2]. Lưu Hồng Dũng, Nguyễn Ánh Việt, Đoàn Thị Bích Ngọc. *"Thuật toán mã hóa - xác thực thông tin phát triển từ mật mã otp"*. Journal of Military Science and Technology, CSCE special issue, 87-93, (2020) (in Vietnamese).

[3]. Luu Hong Dung, Tong Minh Duc, Bui The Truyen. *"A variant of otp cipher with symmetric key solution"*. Journal of Science and Technique - Section on Information and Communication Technology (ICT) - No. 16, (2020). DOI: 10.56651/lqdtu.jst.v9.n02.210.ict.

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

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

[6]. GOST R 34.10-94. Russian Federation Standard. Information Technology. Cryptographic Data security. *"Produce and check procedures of Electronic Digital Signature based on Asymmetric Cryptographic Algorithm"*. Government Committee of the Russia for Standards, (1994) (in Russian).

[7]. J. KATZ, Y. LINDELL. *"Introduction to Modern Cryptography"*. Chapman & Hall/CRC, (2008).

[8]. Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman. *"An Introduction to Mathematical Cryptography"*. ISBN 978-0-387-77993-5. Springer - Verlag, (2008).

[9]. L.C. WASHINGTON. *"Elliptic Curves. Number Theory and Cryptography"*. Chapman & Hall/CRC, (2008).

[10]. D.R. STINSON. *"Cryptography. Theory and Practice"*. Chapman & Hall/CRC, (2006).

[11]. R.A. MOLLIN. *"An Introduction to Cryptography"*. Chapman & Hall/CRC, (2006).

[12]. J. Talbot and D. Welsh. *"Complexity and Cryptography: An Introduction"*. Cambridge University Press, (2006).

[13]. J. H. Silverman. *"Elliptic curves and cryptography"*. In Public-Key Cryptography, volume 62 of Proc. Sympos. Appl. Math., pages 91–112. Amer. Math. Soc., Providence, RI, (2005).

[14]. J. BUCHMANN. *"Introduction to Cryptography"*. Springer–Verlag, (2004).

[15]. W. MAO. *"Modern Cryptography. Theory and Practice"*. Pearson Education, (2004).

[16]. I. SHPARLINSKI. *"Cryptographic Applications of Analytic Number Theory. Complexity Lower Bounds and Pseurandomness"*. Birkhäuser, (2003).

[17]. S.S. WAGSTAFF. *"Cryptanalysis of Number Theoretic Ciphers"*. Chapman & Hall/CRC, (2003).
[18]. I. F. Blake, G. Seroussi, and N. P. Smart. *"Elliptic Curves in Cryptography"*, volume 265 of London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, (2000).
[19]. I. BLAKE, G.SEROUSSI & N. SMART. *"Elliptic Curves in Cryptography"*. Cambridge University Press, (2000).
[20]. A. Menezes. *"Elliptic Curve Public Key Cryptosystems"*. The Kluwer International Series in Engineering and Computer Science, 234. Kluwer Academic Publishers, Boston, (1993).

TÓM TẮT

Phương pháp xây dựng các lược đồ mã khối khóa công khai dựa trên bài toán logarit rời rạc

Bài báo đề xuất phương pháp xây dựng sơ đồ mã khối cho phép xác minh nguồn gốc và tính toàn vẹn của thông điệp được mã hóa. Ngoài ra, khóa bí mật dùng chung giữa người gửi/người mã hóa và người nhận/người giải mã đối với mỗi tin nhắn được mã hóa được thiết lập dựa trên cơ chế mật mã khóa công khai.

Từ khóa: Mật mã khóa đối xứng; Mật mã khóa công khai; Lược đồm mã hóa – xác thực; Bài toán logarit rời rạc; Mật mã OTP.