

# A PUBLIC KEY ENCRYPTION - AUTHENTICATION SCHEME BASED ON ELGAMAL CRYPTOGRAPHIC ALGORITHM

*Kim Thanh Nguyen*<sup>1,\*</sup>, *Minh Thanh Ta*<sup>1</sup>, *Hong Dung Luu*<sup>1</sup>  
DOI: 10.56651/lqdtu.jst.v12.n1.658.ict

## Abstract

The article proposes a public-key cryptographic scheme based on the ElGamal algorithm. This scheme can perform security functions and authenticate the origin and integrity of the encrypted message simultaneously. Moreover, the proposed scheme is also used as the basis for developing an algorithm to establish a shared secret key for symmetric key cryptosystems.

## Index terms

Public key cryptography, symmetric key cryptography, encryption–authentication scheme, discrete logarithm problem.

## 1. Introduction

The ElGamal algorithm [1] is a public key cryptographic algorithm whose security is guaranteed by the difficulty of the discrete logarithm problem. However, because there is no mechanism to verify the origin and integrity of the encrypted message, this algorithm is not resistant to spoofing attacks (such as man-in-the-middle attacks,..). In [2], [3], a variant of the ElGamal algorithm is proposed. In addition to the ability to secure information, these schemes also have the ability to authenticate the origin and integrity of the encrypted message based on the mechanism of digital signatures.

This article proposes a variant of the ElGamal algorithm - this is a type of public key encryption-authentication scheme capable of simultaneously performing two functions of confidentiality and authentication (of origin and integrity) the encrypted message. The authentication of the origin and integrity of the encrypted message without relying on the authentication mechanism of the digital signature is the difference between the

---

<sup>1</sup>Institute of Information and Communication Technology, Le Quy Don Technical University

\*Corresponding author, email: thanhcuohp@gmail.com

scheme proposed here and the schemes in [2], [3]. Here, this scheme is called the ElGamal–style encryption–authentication scheme.

Based on the ElGamal–style encryption–authentication algorithm, this article proposes a public–key block cipher scheme, which can be applied for symmetric–key cryptosystems (such as: DES [4], AES [5],...). In this public–key block cipher scheme, the shared secret key between the sender/encryptor and receiver/decryptor is established based on the mechanism of public–key cryptography for each encrypted message.

## 2. The ElGamal-style encryption-authentication scheme

### 2.1. The proposed scheme

The ElGamal–style encryption–authentication scheme presented in this section includes: the Key Generation algorithm (Algorithm 1), the Encryption algorithm (Algorithm 2) and the Decryption – Authentication algorithm (Algorithm 3), these algorithms are described as follows:

#### 2.1.1. The Parameter and Key Generation algorithm:

---

**Algorithm 1:** Generate parameters and keys

---

**input:**  $l_p, l_q$

**output:**  $p, q, g, y, x$

- 1: Choose a pair of prime numbers  $p, q$  with:  
 $len(p) = l_p, len(q) = l_q$  and  $q|(p - 1)$
  - 2: Choose a value of  $\alpha$  in the range  $(1, p)$ , compute  $g$  according to the formula:  
 $g = \alpha^{\frac{p-1}{q}} \bmod p$ , satisfy  $g \neq 1$
  - 3: Choose a secret key  $x$  in the range  $(1, q)$
  - 4: Calculate the public key  $y$  according to the formula:  
 $y = g^x \bmod p$
- 

*Notes:*

- $len()$  The function that calculates the length (in bits) of an integer.
- $y$ : The public key.
- $x$ : The secret (private) key.
- $p, q, g$ : The system parameters.

Assume  $x_s$  is the secret key of the sender/encryptor and  $x_r$  is the secret key of the receiver/decryptor, then the corresponding public keys of the sender are:

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

And of the receiver is:

$$y_r = g^{x_r} \text{ mod } p$$

### 2.1.2. The Encryption algorithm:

---

**Algorithm 2: Encryption**

---

**input:**  $p, x_s, y_r, P$

**output:**  $(R, C)$

- 1: Compute the value  $S_e$  according to the formula:  
$$S_e = (y_r)^{x_s} \text{ mod } p$$
  - 2: Compute the value  $R$  by:  
$$R = \text{HASH}(P)$$
  - 3: Compute the sender's encryption key  $K_e$  by:  
$$K_e = \text{HASH}(R \parallel S_e)$$
  - 4: Encrypt the plaintext  $P$  according to the formula:  
$$C = P * g^{K_e} \text{ mod } p$$
  - 5: Send ciphertext  $(R, C)$  to the receiver.
- 

#### Notes:

- $y_r$ : The public key of the receiver.
- $x_s$ : The secret (private) key of the sender.
- $P$ : The plaintext.
- $(R, C)$ : The ciphertext corresponding to  $P$ .
- $\text{HASH}()$ : The cryptographic hash function, e.g. *SHA1/SHA256* [6].
- Operator  $\parallel$  is the operation to concatenate two bit strings.

### 2.1.3. The Decryption – Authentication algorithm:

#### Notes:

- $y_s$ : The public key of sender.
- $x_r$ : The secret key of the receiver.
- $M$ : The post-decrypted message.
- $(R, C)$ : The ciphertext corresponding to  $P$ .
- $\text{HASH}()$ : The cryptographic hash function, e.g. *SHA1/SHA256* [6].
- Operator  $\parallel$  is the operation to concatenate two bit strings.

---

**Algorithm 3:** Decryption

---

**input:**  $p, x_r, y_s, (R, C)$

**output:**  $M$

- 1: Compute the value  $S_d$  according to the formula:  

$$S_d = (y_s)^{x_r} \bmod p$$
  - 2: Compute the receiver's decryption key  $K_d$  by:  

$$K_d = \text{HASH}(R \parallel S_d)$$
  - 3: Decrypt the received ciphertext  $C$  according to the formula:  

$$M = C * g^{-K_d} \bmod p$$
  - 4: Compute the value  $V$  by:  

$$V = \text{HASH}(M)$$
  - 5: Checks: if  $V = R$  then the origin and integrity of the post-decrypted message  $M$  is confirmed. Otherwise, if  $V \neq R$ , the validity of the received message will be denied.
- 

**2.2. The correctness of the proposed schema**

What needs to be proved here is: if the received ciphertext is the same as the sent ciphertext, the message after decryption is also the message before encryption:  $M = P$  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:

$$\begin{aligned} S_d &= (y_s)^{x_r} \bmod p = (g^{x_s} \bmod p)^{x_r} \bmod p \\ &= (g^{x_r} \bmod p)^{x_s} \bmod p = y_r^{x_s} \bmod p = S_e \end{aligned}$$

So we have:

$$K_d = \text{HASH}(R, S_d) = \text{HASH}(R, S_e) = K_e$$

Therefore, we have the first proof:

$$\begin{aligned} M &= C * g^{-K_d} \bmod p = (P * g^{K_e} \bmod p) * g^{-K_d} \bmod p \\ &= P * g^{K_e} * g^{-K_d} \bmod p = P * g^{K_e} * g^{-K_e} \bmod p = P \end{aligned}$$

Finally, we have the second proof:

$$V = \text{HASH}(M) = \text{HASH}(P) = R$$

### 2.3. Some evaluation of the security level of the proposed schema

The security level of the proposed schema is evaluated based on its ability to resist some typical attacks as follows:

– *Secret key attack*: To find the receiver's secret key  $x_r$  from the formula:

$$y_r = g^{x_r} \text{ mod } p$$

or the sender's secret key  $x_s$  from:

$$y_s = g^{x_s} \text{ mod } p$$

then the attacker is forced to solve the discrete logarithm problem on a finite field  $Z_p$  [7]–[19]. Currently, no polynomial – time algorithm has been published for this difficult problem.

– *Ciphertext - only attack*: In this case, as well as the above case (Secret key attack), the attacker has only one way to solve the discrete logarithm problem on the finite field to find the sender's secret key or receiver's secret key.

– *Known - plaintext attack*: In this case, in addition to a direct attack on the key Generation algorithm (Algorithm 1) to find the sender's secret key  $x_s$  or receiver's secret key  $x_r$ , the attacker can also calculate the sender's encryption key  $K_e$  from the formula:

$$C = P * g^{K_e} \text{ mod } p$$

or calculate the receiver's decryption key  $K_d$  from:

$$M = C * g^{-K_d} \text{ mod } p$$

then calculate  $S_e$  from:

$$K_e = \text{HASH}(R \parallel S_e)$$

or calculate  $S_d$  from:

$$K_d = \text{HASH}(R \parallel S_d)$$

If the attacker finds  $S_e$  or  $S_d$ , the security of the algorithm is completely broken—similar to the case when the attacker finds the sender's secret key or the receiver's secret key. However, in order to calculate  $K_e$  or  $K_d$  in the above way, the attacker is also forced to solve the discrete logarithm problem on the finite field  $Z_p$ .

– *Spoofing attack*: In the proposed algorithm, the origin and integrity of the message after decryption will be verified if the condition:  $V = R$  is satisfied.

The origin and integrity of the post – decrypted message will be verified if the condition:  $V = R$  is satisfied. From the calculation of the values of  $V$  and  $R$ , the above condition is satisfied only when the following conditions are satisfied:  $S_d = S_e$  and  $M = P$ . Obviously, the condition:  $S_d = S_e$  allows the sender and receiver of the message to verify each other's identities. That also means, the origin of the post–decrypted message is authenticated. The condition  $M = P$  allows the integrity of the message to be verified after decryption.

### 3. The public-key block cipher scheme

#### 3.1. The proposed scheme

The public-key block cipher scheme proposed here is developed based on the scheme in section 2.1. This is also a type of encryption-authentication scheme, which includes: the Parameter and Key Generation algorithm, the Encryption algorithm and the Decryption – Authentication algorithm. The Parameter and Key Generation algorithm of the scheme proposed here is exactly the same as Algorithm 1 of the ElGamal-style encryption-authentication scheme in section 2.1.1. The Encryption algorithm (Algorithm 4) and the Decryption – Authentication algorithm (Algorithm 5) are described as follows:

##### 3.1.1. The Encryption algorithm:

---

#### Algorithm 4: Encryption

---

**input:**  $p, x_s, y_r, P$

**output:**  $(R, C)$

- 1: Compute the value  $S_e$  according to the formula:  

$$S_e = (y_r)^{x_s} \bmod p$$
  - 2: Compute the value  $R$  by:  

$$R = \text{HASH}(P)$$
  - 3: Compute the sender's encryption key  $K_e$  by:  

$$K_e = \text{HASH}(R \parallel S_e)$$
  - 4: Encrypt the plaintext  $P$  according to the formula:  

$$C = E_{K_e}(P)$$
  - 5: Send ciphertext  $(R, C)$  to the receiver.
- 

*Notes:*

- $E_K()$ : The encryption function of a symmetric-key cryptographic algorithm, such as: DES, AES,....
- $(R, C)$ : The ciphertext corresponding to  $P$

##### 3.1.2. The Decryption-Authentication algorithm:

*Notes:*

- $D_K()$ : The Decryption function of a symmetric key cryptographic algorithm, such as: DES, AES,...

#### 3.2. The correctness of the proposed schema

Similar to the ElGamal-style encryption-authentication scheme in section 2.1, what needs to be proved here is: if the received ciphertext is the same as the sent ciphertext, the message after decryption is also the message before encryption:  $M = P$  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

---

**Algorithm 5:** Decryption

---

**input:**  $p, x_r, y_s, (R, C)$

**output:**  $M$

- 1: Compute the value  $S_d$  according to the formula:  

$$S_d = (y_s)^{x_r} \bmod p$$
  - 2: Compute the receiver's decryption key  $K_d$  by:  

$$K_d = \text{HASH}(R \parallel S_d)$$
  - 3: Decrypt the ciphertext  $C$  by  $D_K()$  and  $K_d$ :  

$$M = D_{K_e}(C)$$
  - 4: Compute the value  $V$  by:  

$$V = \text{HASH}(M)$$
  - 5: Checks if:  $V = R$  then the origin and integrity of the post-decrypted message  $M$  is confirmed. Otherwise, if  $V \neq R$ , the validity of the received message will be denied.
- 

message. The correctness of the proposed scheme is proved as follows:

We have:

$$\begin{aligned} S_d &= (y_s)^{x_r} \bmod p = (g^{x_s} \bmod p)^{x_r} \bmod p \\ &= (g^{x_r} \bmod p)^{x_s} \bmod p = (y_r)^{x_s} \bmod p = S_e \end{aligned}$$

So we have:

$$K_d = \text{HASH}(R, S_d) = \text{HASH}(R, S_e) = K_e$$

Therefore, we have the first proof:

$$M = D_{K_d}(C) = D_{K_d}(E_{k_e}(P)) = D_{K_e}(E_{k_e}(P)) = P$$

Finally, we have the second proof:

$$V = \text{HASH}(M) = \text{HASH}(P) = R$$

### 3.3. Some evaluations of the security level of the proposed schema

In the proposed scheme, its confidentiality depends on the security of the symmetric-key cryptographic algorithm used, in addition, it also depends on another important factor that is the establishment of a shared secret key between the sender/encryptor ( $U_s$ ) and the receiver/decryptor ( $U_r$ ).

It is easy to see that in the proposed scheme, the establishment of a shared secret key between  $U_s$  and  $U_r$  is done for each message and has ensured the basic security properties required by a secure key establishment protocol such as:

- *Entity authentication:* This is a property that allows one of the two objects to confirm with certainty the identity of the object participating in the establishment of a shared secret key. In this algorithm, the calculation of  $S_e$  on the sender's side and

$S_d$  on the receiver's side allows the objects participating in the key establishment to completely authenticate each other's identities.

– *Key authentication*: Key authentication, also known as implicit key authentication, is the ability/property that one of two objects ( $U_r$  or  $U_s$ ) can confirm with certainty that there is only the other object ( $U_r$  or  $U_s$ ) can generate a shared secret key. In this algorithm, except  $U_s$  and  $U_r$ , no one can calculate  $S_e$  and  $S_d$  satisfying the condition:  $S_e = S_d$  without solving the discrete logarithm problem on finite field  $Z_p$ .

– *Known key security*: knowing one or more keys shared between  $U_s$  and  $U_r$  also does not allow a third party to compute other keys that have been or will be established by  $U_s$  and  $U_r$ . In this algorithm, in order to calculate  $S_e$  and  $S_d$  from the shared secret keys  $K_e$  and  $K_d$ , the attacker is also forced to solve the discrete logarithm problem on finite field  $Z_p$ .

Finally, the problem of secret key attack and spoofing attack, similar to the ElGamal-style encryption–authentication algorithm in section 2.1, the attacker is also forced to solve the discrete logarithm problem on finite field  $Z_p$ .

## 4. Conclusions

The article proposes a public–key encryption–authentication scheme and a public–key block cipher scheme. The public–key encryption–authentication scheme is capable of simultaneously performing two functions of confidentiality and authentication (of origin and integrity) of the encrypted message. The public–key block cipher scheme is applicable to symmetric–key cryptosystems (such as: DES, AES,..). In this public–key block cipher scheme, the shared secret key between the sender/encryptor and the receiver/decryptor is established based on the mechanism of public–key cryptography for each encrypted message, so it is very suitable for practical applications. However, in order to apply these schemes in practice, a more in-depth assessment of the cryptanalysis, speed and performance of these schemes is required.

## References

- [1] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, jul 1985. doi: 10.1109/TIT.1985.1057074
- [2] L. H. Dũng, “Phát triển thuật toán mật mã khóa công khai dựa trên hệ mật elgamal,” *Các công trình nghiên cứu, phát triển và ứng dụng CNTT&TT*, p. 35, sep 2014. doi: 10.32913/mic-ict-research-vn.v2.n28.133
- [3] N. V. Thai, “An encryption–authentication algorithms developed from the elgamal cryptosystem,” *Journal of Military Science and Technology*, no. 5, pp. 61–70, dec 2021. doi: 10.54939/1859-1043.j.mst.csce5.2021.61-70
- [4] “Federal information processing standards publication: data encryption standard (DES),” Tech. Rep., dec 1993.
- [5] M. J. Dworkin, “Advanced encryption standard (AES),” Tech. Rep., may 2023.
- [6] Q. H. Dang, “Secure hash standard (SHS),” Tech. Rep., Aug 2015.
- [7] L. C. Washington, “Elliptic curves - number theory and cryptography,” may 2003. doi: <https://doi.org/10.4324/9780203484029>
- [8] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 2018.
- [9] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman and Hall/CRC, aug 2007.

- [10] J. Hoffstein, J. Pipher, and J. H. Silverman, *An Introduction to Mathematical Cryptography*. Springer New York, 2014.
- [11] D. R. Stinson, *Cryptography*. Chapman and Hall/CRC, nov 2005.
- [12] R. A. Mollin, *An Introduction to Cryptography*. Chapman and Hall/CRC, sep 2006.
- [13] J. Talbot and D. Welsh, *Complexity and Cryptography*. Cambridge University Press, jan 2006.
- [14] J. H. Silverman, "Elliptic curves and cryptography," pp. 91–112, 2005.
- [15] J. A. Buchmann, *Introduction to Cryptography*. Springer US, 2001.
- [16] W. Mao, *Modern Cryptography. Theory and Practice*. Pearson Education, 2004.
- [17] I. Shparlinski, Ed., *Cryptographic Applications of Analytic Number Theory*. Birkhäuser Basel, 2003.
- [18] S. S. Wagstaff, *Cryptanalysis of Number Theoretic Ciphers*. Chapman and Hall/CRC, aug 2019.
- [19] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*. Cambridge University Press, jul 1999.

Manuscript received 19-03-2023; Accepted 12-06-2023. ■



**Kim Thanh Nguyen** graduated from University of Electronic Information Technology in 2012; Master of Computer Science in 2018 at Xi'an University of Technology, China. Currently, he is a lecturer at the Institute of Information and Communication Technology - Le Quy Don Technical University. Current research directions: Network technology, Cryptography and Information security. E-mail: thanhcuchp@gmail.com



**Minh Thanh Ta** is currently an associate professor and vice dean of Faculty of Information Technology in Le Quy Don Technical University, Vietnam. He is also a Postdoctoral Fellow of the Department of Mathematical and Computing Sciences at Tokyo Institute of Technology. He received his B.S. and M.S in Computer Science from National Defense Academy, Japan, in 2005 and 2008 and his Ph.D. from Tokyo Institute of Technology, Japan, in 2015, respectively. He is a member of IPSJ Japan and IEEE. His research interests lie in the area of watermarking, network security, and computer vision. E-mail: thanhtm@lqdtu.edu.vn



**Hong Dung Luu** graduated from University in Radio Electronics in 1989, Master in Electronics and Communication Engineering in 2000, Doctorate in Electronic Engineering in 2013 from Le Quy Don Technical University. Currently, he is a lecturer at the Institute of Information and Communication Technology - Le Quy Don Technical University. Research field: Cryptography and Information Security. E-mail: luuhongdung@gmail.com

# LƯỢC ĐỒ MÃ HÓA - XÁC THỰC PHÁT TRIỂN TỪ THUẬT TOÁN MẬT MÃ ELGAMAL

*Nguyễn Kim Thanh, Tạ Minh Thanh, Lưu Hồng Dũng*

## **Tóm tắt**

Bài báo đề xuất một lược đồ mã hóa khóa công khai dựa trên thuật toán ElGamal. Lược đồ này có thể đồng thời thực hiện các chức năng bảo mật, xác thực nguồn gốc và tính toàn vẹn của thông điệp được mã hóa. Hơn nữa, lược đồ được đề xuất còn được sử dụng làm cơ sở để phát triển một thuật toán để thiết lập khóa bí mật chia sẻ cho các hệ mã hóa khóa đối xứng.

## **Từ khóa**

Mật mã khóa công khai, mật mã đối xứng, lược đồ mã hóa - xác thực, vấn đề logarithm rời rạc.