An encryption - authentication algorithms developed from the elgamal cryptosystem

Nguyen Vinh Thai[1*], Doan Thi Bich Ngoc[2], Luu Hong Dung[3]

[1]Information Technology Institute, Academy of Military Science and Technology;
[2]University of Information and Communication Technology; Thai Nguyen University;
[3]Department of Information Technology, Military Technical Academy.
*Corresponding author: nguyenvinhthai@gmail.com.
Received 7 September 2021; Revised 5 December 2021; Accepted 15 December 2021.
DOI: https://doi.org/10.54939/1859-1043.j.mst.csce5.2021.61-70

ABSTRACT

The paper proposes encryption - authentication algorithms developed from the Elgamal cryptosystem. There are algorithms included: system parameters, keys, encryption, and authenticated decryption. New proposed algorithms ensure a level of security against attacks: revealing secret keys - compared with RSA, GOST; security - compare with ElGamal; anti-forgery. Simultaneously verify the origin of e-doc and ensure the sender's authentication.

Keywords: Public - Key Cryptography Algorithm; Algorithm cryptographic; Algorithm ciphers ElGamal; Algorithm signcription DSA.

1. INTRODUCTION

Improving the security of public-key cryptographic algorithms is an approach that gets researchers' attention. In this study, also to improve the algorithm's security against some types of attacks, in reality, we developed some encryption - authentication algorithms to perform the functions of information security and authentication of the origin and integrity of the encrypted message simultaneously. Suggested algorithms include: encryption algorithm (Algorithm 3.3); decryption - authentication algorithm (Algorithm 3.4) developed from the ElGamal cryptosystem.

In addition, the signature scheme (Algorithm 3.2) in this proposal also allows end entities (signer) in the same system can use the same set of system parameters (domain parameters) created by the service provider. Therefore, the proposed scheme can be used in building cryptographic protocols against spoofing attacks very effectively.

2. AN ELGAMAL CRYPTOSYSTEM AND APPLICATIONS

2.1. Discrete Logarithm Problem ($DLP_{(p,g)}$)

A pair of positive integers (p, g) where p is a prime number and g be the generator of group Z_p^*. Then, Discrete Logarithm Problem in Z_p or $DLP_{(p,g)}$ is stated as follows:

For every positive integer $y \in Z_p^*$, find x satisfy the following equation:

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

Algorithm for $DLP_{(p,g)}$ can be expressed as an algorithm for function $DLP_{(p,g)}(.)$ with input variable is y, the function value is x of the following equation:

$$x = DLP_{(p,g)}(y)$$

Let be $DLP_{(p,g)}$ is provably hard (is not "natural"), then p and q are positive integers satisfy the following conditions:

$q|(p-1)$: q is the divisor of $(p-1)$.

$ord_p q = g$: order $p \, mod \, q$ is the smallest positive integer g satisfy the equation: $p^g \equiv 1 \, mod \, q$.

$DLP_{(p,g)}$ is hard.

To satisfy the condition $DLP_{(p,g)}$ is hard, at standard FIPS PUB 186 - 4 [1] and Special Publication 800-56A Revision 2 [2] of NIST suggested key pair used in discrete logarithm cryptographic include requirements: $len(p) \geq L; len(q) \geq ord_p q = g$, including L is the minimum length of p and n, N is the minimum length of q (in bit). Follow the safety time shown in table 1 below:

Table 1. *Minimum safety length of DLP.*

Safety time	L (bit)	N (bit)
Up to 2030	2048	224
After 2030	3072	256

$DLP_{(p,g)}$ is the basis for building the ElGamal cryptosystem [5]. Until now, there is no effective algorithm for $DLP_{(p,g)}$, and the security of the algorithm DSA [1] in the digital signature standard (DSS) of the United States is a testament to the difficulty of this problem.

2.2. ElGamal cryptosystem

ElGamal cryptosystem included public key cryptography and digital signature systems, first proposed in 1984. The ElGamal public key cryptography algorithm was then used by National Security Agency, while the digital signature system was developed into DSS [1] of the United States and GOST R34.10-94 [4] of the Russia Federation.

2.2.1. ElGamal public key cryptography algorithm

a) Encryption and Decryption Algorithms

End entities should be to perform the key formation procedure as follows:

Step 1. Choose a prime number large enough so that DLP in Z_p is hard to solve.

Step 2. Choose $g \in Z_p^*$ is a generator of the cyclic.

Step 3. Choose secret key x in an interval $(1, p)$.

Step 4. Compute the public key according to: $y = g^x \, mod \, p$.

Encryption (assume A is a sender with a private/public key (x_A, y_A) and a receiver is B with a private/public key (x_B, y_B)). A performs the following operations to encode and send message M to B, following:

Step 1. Choose a random integer k in the interval $(1, p)$.

Step 2. Compute r according to: $r = g^k \, mod \, p$.

Step 3. Using B's public key, compute: $C = M \times (y_B)^k \, mod \, p$.

Step 4: Send ciphertext including (C, r) to B.

Decryption. B performs the following operations to recover the original message (M) from the received ciphertext (C, r), according to:

Step 1. Compute $U = r^{x_B} \, mod \, p = g^{k.x_B} \, mod \, p$.

Step 2. Compute $U^{-1} = g^{-k.x_B} \, mod \, p$.

Step 3. Recover the original message: $M = C \times U^{-1} \, mod \, p$.

b) Security level

- ElGamal cryptosystem will be broken if a private key x or k can be computed. To compute x or k, one of two discrete logarithm problems needs to be solved, for example: $y = g^x \ mod \ p$ or $R = g^k \ mod \ p$. However, solving this DLP has so far been considered very difficult.

- A weakness that can be attacked in ElGamal cryptosystem is when the value of k is reused. Indeed, Assume that the same value k is used to encrypt the two messages M or M^* into their respective ciphertexts are (C, r) và (C^*, r^*). Then, we will have

$$C \times (C^*)^{-1} \equiv M \ \times \ (g^x)^k \times (M^* \ \times \ (g^x)^k)^{-1} \ mod \ p$$

as a result: $M \times C \times (C^*)^{-1} \equiv M \ mod \ p$

That means, just knowing the content of either message M or M^*, it is easy to know the content of the other message.

The decryption algorithm, it is shown that the implementation of the algorithm does not need the sender/encryptor's information, so it is impossible to know for sure who the sender is and thus is not resistant to spoofing attacks such as "Man-in-the-Middle" or some other types of attacks [3].

2.2.2. ElGamal digital signature algorithm

a) Signing and Verifying

The system parameters and the key of each end entity are formed in the same way as in the ElGamal public key cryptosystem.

Signing

Step 1. Choose a random integer k (sign value) in the interval $(1, p)$.

Step 2. Compute the first component of signature: $r = g^k \ mod \ p$

Bước 3. Compute the second component of signature:

$$s = k^{-1}[H(M) - x \times r] \ mod \ (p - 1)$$

(where $H(.)$ is a hash function)

The signature consists of the pair: (r, s)

Verifying

Step 1. Compute: $w = g^{H(M)} \ mod \ p$

Step 2. Compute: $v = (y^r \times r^s) \ mod \ p$

Checks if $w = v$ then the signature (r, s) is valid, so the origin and integrity of message M are recognized.

b) Security level

Similar to the encryption system, ElGamal digital signature also has a weakness that the value of parameter k cannot be reused. Assuming the value of k is reused when forming a signature on two messages m_1 and m_2, from the equation for computing s of the signature we have:

$$k(s_1 - s_2) \equiv (m_1 - m_2) \ mod \ (p - 1)$$

or

$$k(s_2 - s_1) \equiv (m_2 - m_1)\ mod\ (p-1)$$

as a result

$$k = (s_1 - s_2)^{-1} \times (m_1 - m_2)\ mod\ (p-1)$$

or

$$k = (s_2 - s_1)^{-1} \times (m_2 - m_1)\ mod\ (p-1)$$

Therefore, if the attacker obtains messages m_1, m_2 and the corresponding signatures $(r_1, s_1), (r_2, s_2)$ such that: $gcd(s_1 - s_2, p-1) = 1$ or $gcd(s_2 - s_1, p-1) = 1$ and $gcd(r_1, p-1) = 1$ or $gcd(r_2, p-1) = 1$, then k will be calculated from which the secret key will be calculated by:

$$x = r_1^{-1} \times [k \times s_1 - H(M_1)]\ mod\ (p-1)$$

or

$$x = r_2^{-1} \times [k \times s_2 - H(M_2)]\ mod\ (p-1)$$

In addition, since there is no authentication solution, spoofing attacks are completely possible by an unwanted 3rd party ("Man-in-the-Middle").

A solution to this problem (Man-in-the-Middle attack) is to use a combination of ElGamal encryption algorithm with the US DSA or GOST R34.10-94 of the Russian Federation. Section 2.4 will present combining the ElGamal encryption algorithm with DSA digital signature standard. Completely similar way for the combination of the ElGamal encryption algorithm with the GOST R34.10-94 of the Russian Federation.

2.2.3 The Digital Signature Algorithm

The DSA is an improved version of the ElGamal signature algorithm used in reality applications. The DSA described in FIPS 186-4 includes the following key and parameter generation algorithms, signing algorithms, and verifying algorithms:

a) Domain parameters and keys

Set of system parameters (domain parameters) and keys created by the service provider. Therefore, the proposed scheme can be used in building cryptographic protocols against spoofing attacks very effectively.

The domain parameters and the key of each end entity are algorithmically formed by Certification Authority - CA as follows:

Step 1. Choose prime divisor q, where: $2^{159} < q < 2^{160}$

Step 2. Choose integer t, where $0 \le t \le 8$. And prime number p, where:

$$2^{511+64t} < p < 2^{512+64t} \text{ so that } q|(p-1)$$

Step 3. Choose a random integer $\alpha \in Z_p^*$, compute:

$$g = \alpha^{\frac{p-1}{q}}\ mod\ p, \text{ until } g \neq 1.$$

Step 4. Choose secret key x in an interval $(1, q)$

Step 5. Compute User's Public Key:

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

Step 6. Choose a hash function $H: \{0,1\}^* \to Z_h$ where $q < h < p$

Note:

+ User's Public Key is y, User's Private Key is x.

+ Domain parameters: (p, q, g).

b) Signing algorithm

Signing

Step 1. Choose a random integer k (sign value) in the interval $(1, q)$.

Step 2. Compute $r = \left(g^k \bmod p\right) \bmod q$

Step 3. Compute $s = k^{-1}[H(M) + x \times r] \bmod q$

The signature consists of the pair: (r, s)

Hash function $H: \{0,1\}^* \rightarrow Z_h$ where $q < h < p$

Verifying

Step 1. Discard signature if $0 < s < q$ and $0 < r < q$

Step 2. Compute $w = (s')^{-1} \bmod q$

Step 3. Compute $u_1 = w \times H(M') \bmod q$ and $u_2 = w \times r' \bmod q$

Step 4. Compute $v = [g^{u_1} \times y^{u_2} \bmod p] \bmod q$

Step 5. If $v = r'$ then (r, s) = true, else (r, s) = false

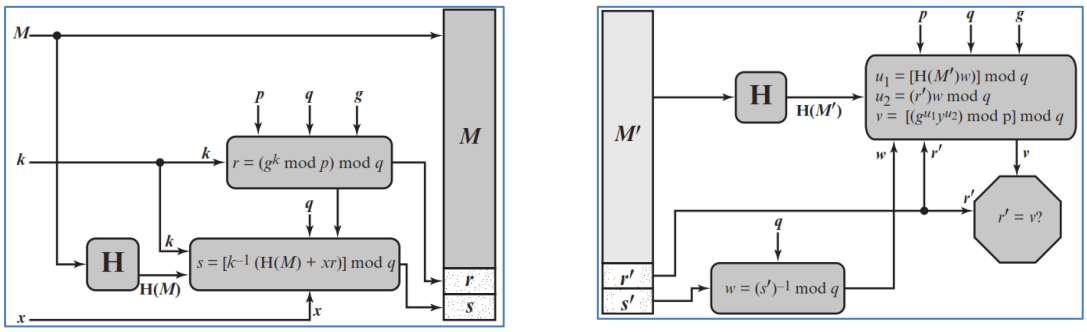
Note:

+ M', r', s': are versions of M, r, s

+ When (r, s) = true: valid signature, message M is authenticated for origin and integrity.

+ When (r, s) = false: signature or/and message is a faked.

The Signing and Verifying process are modeled according to figure 1 below:



a) Signing *b) Verifying*

Figure 1. *DSS Signing and Verifying.*

2.3. ElGamal - DSA Algorithm

The DSA-ElGamal algorithm is developed from the combination of ElGamal cryptographic algorithm and DSA to ensure information security and authentication capabilities. Here information is verified simultaneously for origin as well as for integrity.

In the article *"A method for developing a public key cryptosystem"* [6], the authors have analyzed the correctness and security of ElGamal - DSA in terms of its nature, which is performed (signing/encrypting and signing/decrypting) indirectly through the C message.

ElGamal - DSA Algorithm integrates ElGamal encryption algorithm with a digital signature algorithm to ensure confidentiality and authentication (origin and integrity) of information and can resist real-world spoofing attacks. It is also possible to integrate the ElGamal encryption algorithm with another digital signature algorithm such as GOST R34.10 - 94, Schnorr [5] using the same method.

Thus, integrating the encryption and digital signature (sign then encrypt, sign - encrypt at the same time) has the most significant disadvantage: signing and encryption procedures increase complexity. It can be said that complexity is the total complexity of the encryption and signing procedures.

3. CONSTRUCTING AN ENCRYPTION - AUTHENTICATION ALGORITHMS

Accompanying with the method of integrating digital signature algorithm and encryption algorithm. An encryption-authentication algorithm proposed in the paper will compare safety and efficiency with existing algorithms by the same method (such as ElGamal - DSA, GOST, Schnorr).

3.1. Digital signature scheme

3.1.1. Global Public-Key Components ***(Algorithm 3.1)***

Input: lp, lq - length (bit) of prime numbers p, q.

Output: p, q, g, y, x.

Step 1. Choose prime numbers p, q:

$$len(p) = lp, len(q) = lq \text{ so that } q|(p-1)$$

Step 3. Choose a random integer $\alpha \in Z_p^*$, compute:

$$g = \alpha^{\frac{p-1}{q}} \, mod \, p, \text{ until } g \neq 1.$$

Step 4. Choose secret key x in an interval $(1, q)$.

Step 5. Compute User's Public Key:

$$y = g^{-(x)^{-1}} \, mod \, p \tag{1}$$

Step 6. Choose a hash function $H: \{0,1\}^* \rightarrow Z_h$ where $q < h < p$.

<u>*Note:*</u>

+ $len(.)$: Function that computes the length (in bits) of a number.
+ User's Public Key is y, User's Private Key is x.
+ Domain parameters: (p, q, g).

3.1.2. Digital signature scheme ***(Algorithm 3.2)***

a) Signing

Input: p, q, g, x, M - Message to sign.

Output: (E, S) - The signature.

Step 1. Choose a random integer k in the interval $(1, q)$.

Step 2. Compute

$$R = g^k \, mod \, p \tag{2}$$

Step 3. Compute first component of signature:

$$E = H(M \| R) \, mod \, q \tag{3}$$

Step 4. Compute the second component of signature:

$$S = x \times (E - k) \, mod \, q \tag{4}$$

<u>*Note:*</u>

+ Operator "||" is a concatenation of 2-bit strings.

b) Verifying

Input: p, q, g, y, M - Message to check.

Output: (E, S) = true/false.

Step 1. Compute $\overline{R} = (y)^S \times (g)^E \ mod \ p$.

Step 2. Compute $\overline{E} = H\left(M \| \overline{R}\right)$.

Step 3. If $\overline{E} = E$ then (E, S) = true, else (E, S) = false.

Note:

+ When (E, S) = true: valid signature, message M is authenticated for origin and integrity.

+ When (r, s) = false: signature or/and message is a faked.

3.1.3. The correctness of the proposed new schema

With parameters and keys formed by **Algorithm 3.1**, signature (E, S) created by **Algorithm 3.2** - *a) Signing*, $\overline{E}$ created by **Algorithm 3.2** - *b) Verifying*, then what needs to be proved that: $\overline{E} = E$.

Indeed, due to:

$$\begin{aligned}\overline{R} &= (y)^S \times (g)^E \ mod \ p \\ &= \left(g^{-(x)^{-1}} \ mod \ p\right)^{x \times (E-k) \ mod \ q} \times (g)^E \ mod \ p \\ &= \left(g^{-(x)^{-1}} \ mod \ p\right)^{x \times (E-k)} \times (g)^E \ mod \ p \\ &= (g)^{-(x)^{-1} \times x \times (E-k)} \times (g)^E \ mod \ p \\ &= g^k \ mod \ p = R\end{aligned}$$

As a result:

$$\overline{E} = H\left(M \| \overline{R}\right) = H(M \| R) = E$$

3.2. The performance of the algorithm

The effectiveness of the signature scheme can be assessed by the implementation cost of the signing algorithms, the signature check, and the signature size generated by the scheme. In this section, the effectiveness of the proposed new scheme will be evaluated and compared with the digital signature standards of the United States (DSA) and Russia (GOST R34-10.94).

3.2.1. Signature size

It can be seen that, with the same set of parameters (p, q), the signature size generated by the scheme here with the digital signature standards of the United States and Russia is equivalent.

3.2.2. Cost

Cost of *Signing* and *Verifying* algorithms can be assessed through the number of operations required or the total time required to perform the operations to *Signing* and *Verifying* the signature. Here is the convention of using symbols:

T_{exp}: Time to perform exponential modulo.

T_h: Time to perform a hash function.

T_{mul}: Time to perform multiplication modulo.

T_{inv}: Time to perform the inverse modulo.

Note: The key and parameter generation algorithm needs to be done only once for all end entities. Therefore, the computational cost for the parameter and key generation algorithm can be ignored when computing the digital signature scheme's cost.

The costs for the *Signing* (K), *Verifying* (KT) algorithm of the US, the Russian Federation digital signature standards, and of the proposed new scheme are shown in the following table 2:

Table 2. *Cost of signature schemes.*

	T_{exp}		T_{mul}		T_{inv}		T_h	
	K	KT	K	KT	K	KT	K	KT
DSA	1	2	2	3	1	1	1	1
GOST R34-10.94	1	3	2	3	0	0	1	1
ElGamal - DSA	1	1	1	2	0	1	1	1
New scheme	1	2	1	1	0	0	1	1

3.3. Security level of New scheme

3.3.1. Secret key attack

The key security of the proposed scheme is ensured by the difficulty of solving the discrete logarithmic problem similar to DSA and GOST R34-10.94.

3.3.2. Signature forgery attack

From the conditions of the **Algorithm 3.2** - *b) Verifying*, any pair (E,S) will be considered a valid signature of the object possessing the domain parameters (p, g, y) on message M, if satisfy:

$$E = H(M \| (y)^S \times (g)^E \bmod p) \quad (5)$$

From (5) shows, finding the pair (E, S) by first choosing 1 of 2 values and then computing the remaining value is more difficult than solving DLP. Furthermore, if $H(.)$ is chosen as a highly secure hash function (SHA 256/512,...) then randomly choosing a pair (E, S) satisfying (5) is not feasible at all in reality applications.

3.3.3. Public authentication

With parameters (E, S), any third party using A's public key can authenticate the encrypted signed message.

3.4. Encryption - Authentication algorithm

3.4.1. Encryption algorithm ***(Algorithm 3.3)***

Input: p, q, g, x_A, y_B, M - plaintext.

Output: (C, E, S) - ciphertext.

Step 1. Choose a random integer k in the interval $(1, q)$.

Step 2. Encrypt message M: $C = M \times (y_B)^k \bmod p$.

Where: (x_B, y_B) is the recipient's private/public key (B): $y_B = g^{-(x_B)^{-1}} \bmod p$.

Step 3. Compute

$$R = g^k \bmod p \quad (6)$$

Step 4. Compute first component of signature:

$$E = H(M \| R) \bmod q \quad (7)$$

Step 5. Compute second component of signature:

$$S = x_A \times (E - k)\ mod\ q \quad (8)$$

Ciphertext to be sent include (C, E, S).

Where: (x_A, y_A) is the sender's private/public key (A): $y_A = g^{-(x_A)^{-1}}\ mod\ p$.

3.4.2. Authentication algorithm ***(Algorithm 3.4)***

Input: $p, q, g, x_B, y_A, (C, E, S)$.
Output: M.

Step 1. Compute: $\overline{R} = (y_A)^S \times (g)^E\ mod\ p$

Step 2. Compute: $w = (x_B)^{-1}\ mod\ q$

Step 3. Compute: $U = (\overline{R})^w\ mod\ q$

Step 4. Decryption: $\overline{M} = C \times U\ mod\ p$

Step 5: Compute: $\overline{E} = H(\overline{M} \| \overline{R})$

Step 6. If $\overline{E} = E$ then $M = \overline{M}$, mean, the message received is the message sent by A.

3.4.3. The correctness

We have: $\overline{R} = (y_A)^S \times (g)^E\ mod\ p$

$$= (g)^{-(x_A)^{-1} \times x_A \times (E-k)} \times (g)^E\ mod\ p = g^k\ mod\ p = R$$

and: $w = (x_B)^{-1}\ mod\ q$

deduce: $U = \left(\overline{R}\right)^w\ mod\ q = \left(g^k\ mod\ p\right)^{(x_B)^{-1}} mod\ p = g^{k(x_B)^{-1}} mod\ p$

On the other hand: $C = M \times (y_B)^k\ mod\ p = M \times \left(g^{-(x_B)^{-1}} mod\ p\right)^k mod\ p$

$$= M \times g^{-k.(x_B)^{-1}} mod\ p$$

As a result:

$$\overline{M} = C \times U\ mod\ p = M \times g^{-k.(x_B)^{-1}} \times g^{k(x_B)^{-1}} mod\ p = M$$

3.4.4. Safety and effective

Similar analysis as in section 3.2 shows that the safety level of the proposed new algorithm is equivalent to that of DSA-ElGamal algorithm, but the advantages of the new algorithm proposed here compared to the combined algorithm ElGamal with DSA (or ElGamal combined with GOST R34.10-94, etc.) is in the performance of the algorithm. The comparison of the efficiency shows that the proposed new algorithm's performance efficiency is higher than the combination of ElGamal with DSA (or ElGamal combined with GOST R34.10-94, etc.).

4. CONCLUSIONS

The article proposes to build an encryption-authentication algorithm of a public-key cryptosystem from the development of the ElGamal encryption algorithm. The algorithm proposed here has the same security features as ElGamal's algorithm. Moreover, they also have a mechanism to verify the origin and integrity of the encrypted message, so it can effectively resist all kinds of attacks-known forgery. Besides, the theoretical comparison confirms that the performance of the proposed algorithm is higher than that of RSA, GOST, ElGamal - DSA methods. Hence, it is suitable for applications with safety requirements and high speed in reality.

REFERENCES

[1]. National Institute of Standards and Technology, FIPS PUB 186-4, July 2013.

[2]. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf.

[3]. Mark Stamp, Richard M. Low. *"Applicd cryptanalysis: Breaking Ciphers in the Real World".* John Wiley & Sons, Inc., ISBN 978-0-470-1.

[4]. 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).

[5]. *"Cryptography and Network Security: Principles and Practice",* 7th Edition, ISBN 978-0-13-444428-4, by William Stallings 2017.

[6]. Nguyễn Lương Bình, Lưu Hồng Dũng, Tống Minh Đức (8-2018), "*Một phương pháp phát triển hệ mật khóa công khai*", *Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XI về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR)*, Hà Nội, ngày 09-10/8/2018. DOI: 10.15625/vap.2018.00042 (in Vietnamese).

TÓM TẮT

THUẬT TOÁN MÃ HÓA - XÁC THỰC PHÁT TRIỂN TỪ HỆ MẬT ELGAMAL

Bài báo đề xuất xây dựng thuật toán mã hóa - xác thực phát triển từ hệ mật ElGamal. Ở đó, bao gồm các thuật toán: hình thành tham số hệ thống, khóa; mã hóa và giải mã có xác thực. Các thuật toán đề xuất mới đảm bảo mức độ an toàn trước các tấn công: làm lộ khóa bí mật - so sánh với RSA, GOST; tính bảo mật - so sánh với ElGamal; khả năng chống giả mạo. Đồng thời xác thực nguồn gốc văn bản điện tử, cũng như đảm bảo việc xác thực người gửi.

Từ khóa: Public - Key Cryptography Algorithm; Algorithm cryptographic; Algorithm ciphers ElGamal; Algorithm signcription DSA.