International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 12, Issue 6 [June. 2023] PP: 36-41

A Type Of Encryption-Authentication Scheme Based On The Elgamal Cryptographic Algorithm

Ho Kim Giau 1[1]*, Luu Hong Dung 2[2]
[1]*Telecommunications University*
[2] *Le Quy Don Technical University*

ABSTRACT
This paper proposes an encryption-authentication scheme based on the ElGamal cryptographic algorithm. This proposal can perform the two functions of security and authentication of encrypted messages simultaneously

KEYWORDS
Authentication
ElGamal
Encryption
Encryption-authentication scheme
Public–key cryptography

Date of Submission: 26-05-2023 Date of acceptance: 07-06-2023

I. Introduction

In [1,2], a type of encryption–authentication scheme is proposed based on the ElGamal public–key cryptographic algorithm [3]. In addition to the ability to secure information, this scheme also has the ability to authenticate the origin and integrity of the encrypted message based on the mechanism of digital signatures.

This paper proposes a type of encryption–authentication scheme which is a variant of the ElGamal cryptographic algorithm, which can simultaneously perform two functions of confidentiality and authentication of encrypted messages. 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 scheme proposed here and the scheme in [1,2].

II. The Encryption – Authentication scheme

2.1. The Encryption – Authentication scheme on the finite filed

The Encryption – Authentication scheme proposed here includes: the Parameter and Key Generation algorithm (**Algorithm 1.1**), the Encryption algorithm (**Algorithm 1.2**) and the Decryption – Authentication algorithm (**Algorithm 1.3**), described as follows:

2.1.1. The Parameter and Key Generation algorithm

Algorithm 1.1:
input: $\boldsymbol{l_p}$, $\boldsymbol{l_q}$.
output: p, q, g, y, x.
[1]. Choose a pair of prime numbers **p**, **q** with: *len*(p) = $\boldsymbol{l_p}$, *len*(q) = $\boldsymbol{l_q}$ and: q|(p-1).
[2]. Choose a value of α in the range (1,p), compute **g** according to the formula:
g = $\alpha^{\frac{p-1}{q}}$ mod p, satisfy g ≠ 1.
[3]. Choose a secret key **x** in the range (1,q).
[4]. Compute the public key **y** according to the formula:
y = g^x mod p

<u>Notes</u>:
– *len()*: the function that calculates the length (in bits) of an integer.
– x: the secret (private); **y**: the public key.
– p, q, g: the system parameters.

Assume $\mathbf{x_s}$ is the secret key of the sender (encryptor) and $\mathbf{x_r}$ is the secret key of the receiver (decryptor), then the corresponding public keys of the sender are:
$y_s = g^{xs}$ mod p

* Corresponding author. *Email: hkgiau@gmail.com*

and of the receiver is:
$y_r = g^{xr} \bmod p$

2.1.2. The Encryption algorithm

Algorithm 1.2:
input: p, x_s , y_r , P.
output: (R,C).
[1]. Compute the value S_e according to the formula:
$S_e = (y_r)^{xs} \bmod p$
[2]. Compute the value K_s by:
$K_s = HASH(P,S_e)$
[3]. Compute the value R according to the formula:
$R = g^{Ks} \bmod p$
[4]. Encrypt the plaintext P according to the formula:
$C = P.(y_r)^{Ks} \bmod p$
[5]. Send ciphertext (R,C) to the receiver.

Notes:
– x_s: the secret key of the sender.
– y_r: the public key of the receiver.
– P: the plaintext.
– (R,C): the ciphertext corresponding to P.
– HASH(): The cryptographic hash function, e.g. SHA1/SHA256 [4],...

2.1.3. The Decryption – Authentication algorithm

Algorithm 1.3:
input: p, x_r , y_s , (R,C).
output: M.
[1]. Decrypt the received message C according to the formula:
$M = C.R^{-xr} \bmod p$
[2]. Compute the value S_d according to the formula:
$S_d = (y_s)^{xr} \bmod p$
[3]. Compute the value K_r by:
$K_r = HASH(M,S_d)$
[4]. Compute the value V according to the formula:
$V = g^{Kr} \bmod p$
[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 thereceived message will be denied.

Notes:
– x_r: the secret key of the receiver.
– y_s: the public key of sender.
– M: the post–decrypted message.

2.1.4. The correctness of the proposed schema

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 = 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:
$S_d = (y_s)^{xr} \bmod p = (g^{xs} \bmod p)^{xr} \bmod p$
$= (g^{xr} \bmod p)^{xs} \bmod p = (y_r)^{xs} \bmod p = S_e$
Therefore, we have the first proof:
$M = C.R^{-xr} \bmod p = (P.(y_r)^{Ks} \bmod p) . (g^{Ks} \bmod p)^{-xr} \bmod p$
$= P.(g^{xr} \bmod p)^{Ks} . (g^{Ks} \bmod p)^{-xr} \bmod p$
$= P . g^{xr . Ks} . g^{-xr . Ks} \bmod p = P$
So we have:
$K_r = HASH(M,S_d) = HASH(P,S_e) = K_s$
Then, we have the second proof:
$V = g^{Kr} \bmod p = g^{Ks} \bmod p = R$

2.1.5. Some evaluation of the security level of the proposed schema

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

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

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

or the sender's secret key x_s from:

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

then the attacker is forced to solve the discrete logarithm problem on finite field Z_p [8-21]. Currently, no polynomial–time algorithm has been published for this hard 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 1.1**) 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_s from the formula:

$C = P.(y_r)^{Ks} \bmod p$

or from the formula:

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

then calculate S_e from:

$K_s = HASH(P,S_e)$

If the attacker finds S_e, the security of the scheme 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_s in the above way, the attacker is also forced to solve the discrete logarithm problem on the finite field Z_p.

In addition, an attacker can find the receiver's secret key x_r from solving the equation:

$M = C.R^{-xr} \bmod p$

However, in this case the attacker still needs to solve the discrete logarithm problem on Z_p.

– *Spoofing attack*: In the proposed scheme, 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.

2.2. The encryption – authentication scheme on the elliptic curve

The Encryption – Authentication scheme proposed here includes: the Key Generation algorithm (**Algorithm 2.1**), the Encryption algorithm (**Algorithm 2.2**) and the Decryption – Authentication algorithm (**Algorithm 2.3**), described as follows:

2.2.1. The Key Generation algorithm

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

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

$$\mathbf{y^2 = x^3 + ax + b}$$

with: $\mathbf{a,b \in F_p}$ and satisfied: $\mathbf{4a^3 + 27b^2 \neq 0 \bmod q}$

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

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 Generation Algorithm is described as follows:

```
Algorithm 2.1:
input: E(F_p) = (p,a,b,G,q).
output: (d,P).
[1]. Generate the secret key d in the range (1,q):
d = PRNG({1,2,…,q –1})
[2]. Calculate the public key P according to the formula:
P = (x_p,y_p) = d . G
```

Notes:

– PRNG(): The Random/Pseudo-random number generator.

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

Assume $\mathbf{d_s}$ is the secret key of the sender (encryptor) and $\mathbf{d_r}$ 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:

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

2.2.2 The Encryption algorithm

Algorithm 2.2:

input: $E(F_p) = (p,a,b,G,q)$, d_s, P_r, m_1.

output: (R,C).

[1]. Represent the message $\mathbf{m_1}$ as a point M_1 on $E(F_p)$.

[2]. Compute the $\mathbf{S_e}$ according to the formula:

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

[3]. Compute the value K_s by:

$K_s = HASH(m_1,x_{se})$

[4]. Compute the R according to the formula:

$R = K_s . G$

[5]. Encrypt the message $\mathbf{m_1}$ according to the formula:

$C = M_1 + K_s . P_r$

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

Notes:

– (x_{se},y_{se}): The coordinates of the point $\mathbf{S_e}$ on $E(F_p)$.

– m_1: The plaintext.

– (R,C): The ciphertext corresponding to $\mathbf{m_1}$.

– HASH(): The cryptographic hash function, e.g. SHA1/SHA256 [4]…

2.2.3. The Decryption – Authentication algorithm

Algorithm 2.3:

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

output: m_2.

[1]. Decrypt the received message C according to the formula:

$M_2 = C – d_r . R$

[2]. Convert M_2 to the corresponding message $\mathbf{m_2}$.

[3]. Compute the $\mathbf{S_d}$ according to the formula:

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

[4]. Compute the value K_r by:

$K_r = HASH(m_2,x_{sd})$

[5]. Compute the V according to the formula:

$V = K_r . G$

[6]. Check if: V = R then the origin and integrity of the post–decrypted message $\mathbf{m_2}$ is confirmed. Otherwise, if $V \neq R$, the validity of the received message will be denied.

Notes:

– (x_{sd},y_{sd}): The coordinates of the point S_d on $E(F_p)$.

– m_2: The post–decrypted message.

2.2.4. The correctness of the proposed schema

Similar to the Encryption–Authentication scheme in **Section 1**, 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: $\mathbf{m_2 = m_1}$ 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. The correctness of the proposed scheme is proved as follows:

We have:

$$M_2 = C – d_r . R = M_1 + K_s . P_r – d_r . R$$
$$= M_1 + K_s . d_r . G – d_r . K_s . G = M_1$$

Therefore, we have the first proof: $m_2 = m_1$

We also have:

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

$= d_s . P_r = S_e$

So we have:

$K_r = HASH(m_2,x_{sd}) = HASH(m_1,x_{se}) = K_s$

Then, we have the second proof:

$V = K_r . G = K_s . G = R$

2.2.5. Some evaluation of the security level of the proposed schema

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

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

$P_r = d_r . G$

or the sender's secret key d_s from:

$P_s = d_s . G$

then the attacker is forced to solve the discrete logarithm problem on $E(F_p)$ [8-21]. Currently, no polynomial–time algorithm has been published for this hard 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 $E(F_p)$ 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 2.1**) to find the sender's secret key d_s or receiver's secret key d_r, the attacker can also calculate the sender's encryption key K_s from the formula:

$R = K_s . G$

or from the formula:

$C = M_1 + K_s . P_r$

then calculate x_{se} from:

$K_s = HASH(m_1,x_{se})$

If the attacker finds x_{se}, 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 x_{se} in the above way, the attacker is also forced to solve the discrete logarithm problem on $E(F_p)$.

In addition, an attacker can find the receiver's secret key d_r from solving the equation:

$M_2 = C – d_r . R$

However, in this case the attacker still needs to solve the discrete logarithm problem on $E(F_p)$.

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

The origin and integrity of the post–decrypted message will be verified if the condition: $\mathbf{V = R}$ is satisfied. From the calculation of the values of **V** and **R**, the above condition is satisfied only when the following conditions are satisfied: $\mathbf{S_d = S_e}$ and $\mathbf{m_2 = m_1}$. Obviously, the condition: $\mathbf{S_d = S_e}$ 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 $\mathbf{m_2 = m_1}$ allows the integrity of the message to be verified after decryption.

III. Conclusion

The paper proposes a type of encryption – authentication scheme which is a variant of the ElGamal public–key cryptographic logarithm. Since there is a mechanism to authenticate of the origin and integrity of the encrypted message, the scheme proposed here is resistant to various types of spoofing attacks, which is one of the basic requirements by practical applications

REFERENCES

[1]. Luu Hong Dung. Development of a public key cryptographic algorithm based on the Elgamal cryptography. Research, development and application of ICT, V-1, No 8 (28), 12/2012. DOI: https://doi.org/10.32913/mic-ict-research-vn.v2.n28.133.

[2]. Nguyen Vinh Thai, Doan Thi Bich Ngoc, Luu Hong Dung. An encryption - authentication algorithms developed from the elgamal cryptosystem. Journal of Military Science and Technology, Special issue No.5, 12 - 2021. ISSN: 1859-1043. DOI: https://doi.org/10.54939/1859-1043.j.mst.csce5.2021.61-70.

[3]. Elgamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE TRANSACTIONS ON INFORMATION THEORY (1985).

[4]. National Institute of Standards and Technology, NIST FIPS PUB 180-1. April 1995.

[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]. A. Menezes, P. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography. CRC Press.

[9]. J. KATZ, Y. LINDELL. Introduction to Modern Cryptography. Chapman & Hall/CRC 2008.

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

[11]. L.C. WASHINGTON. Elliptic Curves. Number Theory and Cryptography. Chapman & Hall/CRC 2008.

[12]. D.R. STINSON. Cryptography. Theory and Practice. Chapman & Hall/CRC 2006.

[13]. R.A. MOLLIN. An Introduction to Cryptography. Chapman & Hall/CRC 2006.
[14]. J. Talbot and D. Welsh. Complexity and Cryptography: An Introduction. Cambridge University Press, 2006.
[15]. 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.
[16]. J. BUCHMANN. Introduction to Cryptography. Springer–Verlag 2004.
[17]. W. MAO. Modern Cryptography. Theory and Practice. Pearson Education 2004.
[18]. I. SHPARLINSKI. Cryptographic Applications of Analytic Number Theory. Complexity Lower Bounds and Pseurandomness. Birkhäuser 2003.
[19]. S.S. WAGSTAFF. Cryptanalysis of Number Theoretic Ciphers. Chapman & Hall /CRC 2003.
[20]. 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.
[21]. I. BLAKE, G.SEROUSSI & N. SMART. Elliptic Curves in Cryptography. Cambridge University Press 2000.