A HIGH-PERFORMANCE AUTHENTICATION-ENCRYPTION SCHEME

Kim Thanh Nguyen[1], *Minh Thanh Ta*[1], *Hong Dung Luu*[1]

https://doi.org/10.56651/lqdtu.jst.v11.n01.360.ict

Abstract

This paper proposes an encryption scheme that enables verifying the source and integrity of the encrypted message. Additionally, the proposed scheme generates a secret key which is shared between the sender/encryptor and the receiver/decryptor for each encrypted message using public key cryptography.

Index terms

Symmetric key cryptography, public key cryptography, encryption – authentication, hash function, discrete logarithm.

1. Introduction

In [1], a solution was proposed for building a symmetric-key cryptosystem using OTP or Vernam ciphers [2]. The benefit of the algorithms constructed in accordance with [1] is that they inherit the security and efficiency of the OTP cipher, 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 technique based on the solution in [1] that includes encryption and authentication of the source as well as the integrity of the encrypted message. On the other hand, the scheme enables the establishment of a secret key sender/encryption-receiver/decryption sharing mechanism similar to that used in signcryption schemes [3], [4], [5], [6]. Additionally, since the proposed technique does not rely on digital signatures to authenticate encrypted data, it performs better than earlier signcryption systems.

2. Authentication-encryption scheme

2.1. Proposed scheme

The proposed authentication-encryption scheme is fundamentally similar to the signcryption systems [3], [4], [5], [6], with the assumption that no digital signature is required to authenticate the received message. The scheme includes algorithms for key and parameter generation, encryption, and decryption, described as follows:

[1]Faculty of Information Technology, Le Quy Don Technical University

2.1.1. 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: $len(p) = l_p$, $len(q) = l_q$, $q|(p-1)$
2 Choose $\alpha$ in the range $(1, p)$, calculate $g$ by: $g = \alpha^{\frac{p-1}{q}} \mod p$, satisfy $g \neq 1$
3 Select a secret key $x$ in the range $(1, q)$
4 Calculate the public key by: $y = g^x \mod p$
5 Choose hash function $F_1 : \{0,1\}^* \to Z_h$ with: $q < h < p$
```

Notes:

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

2.1.2. Encryption algorithm:

```
Algorithm 2: Encryption
  input: $g, p, q, x_s, y_r, P$
  output: $(R, C)$
1 Calculate the value of $S_e$ according to the formula: $S_e = (y_r)^{x_s} \mod p$
2 Calculate $R$ according to: $R = F_1(P, S_e)$
3 Calculate the value of the encryption key $K_e$ (shared secret key of the sender)
    by: $K_e = F_1(R, S_e)$
4 Use the encryption function with the symmetric key $E_K()$ to encrypt the
    message to be sent $P : C = E_{K_e}(P)$
5 Send ciphertext $(R, C)$ to the receiver
```

Notes:

- x_s is the secret key of the sender.
- y_r is the public key of the receiver. $(y_r = g^{x_r} \mod p)$.
- P is the plaintext.
- (R, C) are the ciphertext. corresponding to P.

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

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

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

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

Single-use key K_{OT} consists of n subkeys K_i whose size corresponds to the size of the plaintext block :

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

The encryption function $E_K()$ is described as follows:

Algorithm 3: Encryption function with symmetric key

```
  input: $P = \{P_1, P_2, \ldots, P_n\}, K_e$
  output: $C = \{C_1, C_2, \ldots, C_n\}$
1 $K_1 = K_e$
2 for $i = 1 \to n$ do
3     $C_i = P_i \bigoplus K_i$
4     $K_{i+1} = F_2(P_i, K_i)$
5 end
6 Return $C$
```

Notes:

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

2.1.3. Decryption algorithm:

Algorithm 4: Decryption – authentication

```
  input: $p, q, g, x_r, y_s, (R, C)$
  output: $M$
1 Calculate the value of $S_d$ according to: $S_d = (y_s)^{x_r} \mod p$
2 Calculate the value of decryption key $K_d$ (shared secret key of the receiver)
    according to: $K_d = F_1(R, S_d)$
3 Use the decryption function with the symmetric key $D_K()$ to decrypt $C$:
    $M = D_{K_d}(C)$
4 Calculate the value of $V$ according to: $V = F_1(M, S_d)$
5 Check if: $V = R$ then: $M = P$ or the source and integrity of the post-decrypted
    message is confirmed
```

Notes:

- x_r is the secret key of the receiver.

- y_s is the public key or sender. ($y_s = g^{x_s} \mod p$).

The decryption function with the symmetric key $D_K()$ is constructed on [1] 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:

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

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

$$K_{OT} = \{K_1, K_2, \ldots, K_n\}.with : (K_1 = K_d)$$

The decryption function $D_K()$ then has the form:

Algorithm 5: Decryption function with symmetric key

```
   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
3      $M_i = C_i \bigoplus K_i$
4      $K_{i+1} = F_2(M_i, K_i)$
5  end
6  Return $M$
```

2.1.4. The correctness of the proposed scheme:

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:

$$S_d = (y_s)^{x_r} \mod p = (g^{x_s} \mod p)^{x_r} \mod p$$
$$= (g^{x_r} \mod p)^{x_s} \mod p = (y_r)^{x_s} \mod p = S_e$$

So:

$$K_d = F_1(R, S_d) = F1(R, S_e) = K_e$$

Therefore, we have the first proof:

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

Then, we have the second proof:

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

2.2. An application scheme

An application implementation of the proposed new scheme is to use the SHA–256 hash function [7] to perform the roles of functions F_1 and F_2 as suggested in [8]. In this scheme, the plaintext P is encrypted as n data blocks of size 256-bits:

$$P = \{P_1, P_2, \ldots, P_i, \ldots, P_n\}, i = 1 \rightarrow n, |P_i| = 256bits$$

The sent ciphertext consists of two components R and C. The size of component R is a 256-bits prime number p, and the C_4 includes n 256-bits data blocks:

$$C = \{C_1, C_2, \ldots, C_i, \ldots, C_n\}, i = 1 \rightarrow n, |C_i| = 256bits$$

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

$$K_{OT} = \{K_1, K_2, \ldots, K_i, \ldots, K_n\}, i = 1 \rightarrow n, |K_i| = 256bits$$

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

$$M = \{M_1, M_2, \ldots, M_i, \ldots M_n\}, i = 1 \rightarrow n, |M_i| = 256bits$$

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

Algorithm 6: Encryption

```
  input: $g, p, q, x_s, y_r, P$
  output: $(R, C)$
1 Calculate the value of $S_e$ according to: $S_e = (y_r)^{x_s} \mod p$
2 Calculate the value $R$ according to: $R = SHA256(P \,||\, S_e)$
3 Calculate the encryption key $K_e$ according to: $K_e = SHA256(R \,||\, S_e)$
4 $K_1 = K_e$
5 for $i = 1 \rightarrow n$ do
6     $C_i = P_i \bigoplus K_i$
7     $K_{i+1} = SHA256(P_i \,||\, K_i)$
8 end
9 Send ciphertext $(R, C)$ to the receiver
```

Algorithm 7: Decryption – authentication

```
  input: $g, p, q, x_r, y_s, R, C$
  output: $M$
1 Calculate the value of $S_d$ according to: $S_d = (y_s)^{x_r} \mod p$
2 Calculate the value of the decryption key $K_d$: $K_d = SHA256(R \,||\, S_d)$
3 $K_1 = K_d$
4 for $i = 1 \to n$ do
5     $M_i = C_i \bigoplus K_i$
6     $K_{i+1} = SHA256(M_i \,||\, K_i)$
7 end
8 Calculate the value of $V$ according to: $V = SHA256(M \,||\, S_d)$
9 Check if: $V = R$ then return the result: $M = \{M_1, M_2, , \ldots, M_n\}$. Otherwise,
   if: $V \neq R$ then: return $M = \{0, 0, \ldots, 0\}$
```

Note:

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

2.3. Evaluation of the proposed new scheme

The implementation of scheme evaluation focus on two aspects: security level and performance efficiency.

2.3.1. Security level:

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

- *Ciphertext-only Attack (COA)*: the analysis in [1], [8] have demonstrated that a direct attack on the symmetric key ciphers $\{E_K(), D_K()\}$ is not viable when only ciphertext is available. With the proposed scheme, the attacker has the ability to attack the shared secret key (K_e, K_d) using the public parameters p, g, y_s, y_r as well as the R component of the ciphertext. At the first stage, the attacker must find out the sender's secret key x_s in order to compute S_e as following formula:

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

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

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

Subsequently, the shared-secret-key of the sender can be calculated by the formula: $K_e = F_1(R, S_e)$, or the shared-secret-key of the receiver can be figured out as follows:

$K_d = F_1(R, S_d)$. However to calculate x_s from:

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

or x_r from:

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

The attacker needs to solve the discrete logarithm problem – DLP (Discrete Logarithm Problem) [7]. Currently, no polynomial-time algorithm has been published for this difficult problem.

- *Known-plaintext attack(KPA):* Calculating K_e or K_d makes no sense in this scenario, since this key is only utilized once. However, the attacker can still calculate K_e or K_d for subsequent encryption sessions using S_e or S_d. The attacker may then use the public message M to compute S_e using $R = F_1(P, S_e)$ or $K_e = K_1 = M_1 \bigoplus C_1$, and then calculate Se using $K_e = F_1(R, S_e)$. In either case, the attacker will be unable to accomplish his purpose owing to the $F_1()$ hash's one-way character. By deriving S_d from $K_d = F_1(R, S_d)$, the attacker faces the identical challenges as described before.

- *Spoofing attack:* In the proposed technique, an attacker attempting to impersonate a particular sender in order to transmit a forged message to the receiver must access the secret parameters S_e or S_d of the sender. However, based on the analysis above, it is not viable if the attacker cannot solve the DLP problem or the one-way problem of the hash-function. On the other hand, when the following constraints are satisfied, the decrypted message is validated for its source and integrity: $F_1(M, S_d) = F_1(P, S_e)$. Due to collision resistance of F_1, satisfying the preceding condition implies the fulfillment of the following two conditions: $M = P$ and: $S_d = S_e$. With $M = P$, the receiver may verify the message's integrity entirely after decryption, and the message's source is validated using the condition: $S_d = S_e$ in the following manner: because the decryptor uses the public key y of the sender to produce S_d, as follows:

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

In order to $S_d = S_e$, S_e must be generated from the secret key of the sender x_s depended on the formula:

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

Only the owner of the public key y_s knows the corresponding secret key x_s, i.e. only the owner of the public key y_s is capable of generating S_e equal to S_d of the receiver, allowing the receiver to verify that the source of the decrypted message was generated by the owner of the public key y_s. When an attacker sends a spoofed message to a receiver using a value other than the impostor's key x_s (because the attacker does not know x_s), the value S_d generated by the receiver will differ from the impostor's S_e, resulting in the message being rejected, even if the message is meaningful [1].

2.3.2. Efficiency:

The implementation efficiency of the proposed scheme is basically evaluated by comparing the implementation cost of it with the implementation cost of the signcryption ones built on the finite field Z_p in [3], [4], [5], [6]. The comparison with the signcryption schemes built on the elliptic curve (EC) like [9], [10] will be implemented with an EC-installed version of the proposed scheme and will be presented in another paper.

Implementation cost or computation cost is the number of operations to be performed or the total time required to perform the operations of the scheme, here the convention uses the notation:

- T_{exp}: execution time of the exponential modulo operation.
- T_h: execution time of the hash function.
- T_{mul}: execution time of the multiplication modulo.
- T_{inv}: execution time of the modulo inverse.
- T_{Ek}: execution time of the encryption function $E_K()$.
- T_{Ek}:execution time of the decryption function $D_K()$.

Note: The algorithm for generating parameters and keys only needs to be done once for each scheme. Therefore, in comparing the implementation costs of those schemes, the computation cost of the keys and parameters generation algorithms can be ignored.

Implementation costs for the encryption algorithms and the decryption algorithms of the schemes in [3], [4], [5], [6] and the suggested scheme are shown in Table 1 and Table 2 as follows:

Table 1. Implementation cost of encryption algorithms

	T_{exp}	T_{mul}	T_{inv}	T_h	E_k
Zheng Scheme	1	0	1	2	1
Bao – Deng Scheme	2	0	1	2	1
Ma – Chen Scheme	2	1	0	2	1
Savu Scheme	1	1	0	2	1
MTA Scheme	1	0	0	2	1

Table 2. Implementation cost of encryption algorithms

	T_{exp}	T_{mul}	T_{inv}	T_h	D_k
Zheng Scheme	2	2	0	2	1
Bao – Deng Scheme	3	1	0	2	1
Ma – Chen Scheme	3	1	1	2	1
Savu Scheme	3	1	1	2	1
MTA Scheme	1	0	0	2	1

Notes:

- Signcryption schemes Zheng, Bao - Deng, Ma – Chen, and Savu have been proposed in [3], [4], [5], [6].
- MTA scheme is the proposed scheme.

Comment 1: Comparing the implementation cost of the proposed scheme(MTA) with the signcryption schemes [3], [4], [5], [6] as shown in Table 1 and Table 2 shows the better performance of the proposed scheme.

The second aspect of the performance is the number of components and the size of the ciphertext that the schemes produce, as shown in Table 3 as follows:

Table 3. Components and size of the generated ciphertext

	The Components	K ciphertext size
Zheng Scheme	C, R, S	$\lvert P\rvert + \lvert H\rvert + \lvert q\rvert$
Bao – Deng Scheme	C, R, S	$\lvert P\rvert + \lvert H\rvert + \lvert q\rvert$
Ma – Chen Scheme	C, R, S	$\lvert P\rvert + \lvert H\rvert + \lvert q\rvert$
Savu Scheme	C, R, S	$\lvert P\rvert + \lvert H\rvert + \lvert q\rvert$
MTA Scheme	$C, CHEAP$	$\lvert P\rvert + \lvert H\rvert$

Comment 2: The comparison of the number of components and the size of the ciphertext generated by the schemes also shows that the efficiency of the proposed scheme is higher than the existing signcryption schemes [3], [4], [5], [6].

In addition, the execution time of the functions $D_k()$, $E_k()$ built according to [1] is also an important factor to significantly improve the performance of the proposed scheme compared with other signcryption schemes.

3. Conclusion

The article proposes a solution for constructing an encryption-authentication scheme using OTP and public key cryptography. The advantage of encryption schemes based on this technique is that although the security and efficiency of the OTP are preserved, the shared secret key may be used to encrypt each message uniquely according to the encryption mechanism. These are very important properties for these schemes to be applicable in practice. Additionally, because of the process for authenticating the source and integrity of the encrypted message, these encryption systems are resistant to spoofing attacks, which is one of the fundamental requirements for real-world applications.

References

[1] L. H. Dung and N. A. Viet, "A solution to build a symmetric-key cryptosystem," *Information Security Magazine*, vol. 057, no. 5, 2020.

[2] G. Vernam, "US patent 1,310,719," July 1919.

[3] Y. Zheng, "Digital signcryption or how to achieve cost(signature &amp encryption) $\ll$ cost(signature) cost(encryption)," pp. 165–179, 1997.

[4] F. Bao and R. H. Deng, "A signcryption scheme with signature directly verifiable by public key," pp. 55–59, 1998.

[5] G. Feng, F. Bao, C. Ma, and K. Chen, "Efficient authenticated encryption schemes with public verifiability."

[6] L. Savu, "Signcryption scheme based on SCHNORR digital signature," *International Journal of Peer to Peer Networks*, vol. 3, no. 1, pp. 139–10, Jan 2012.

[7] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[8] L. H. Dung, T. M. Duc, and B. T. Truyen, "Variant of OTP cipher with symmetric-key solution," *Journal of Science and Technique - Le Quy Don Technical University*, p. 213, December 2020.

[9] S. A. Ch, N. uddin, M. Sher, A. Ghani, H. Naqvi, and A. Irshad, "An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 74, no. 5, pp. 1711–1723, Oct 2014.

[10] V. Rajasekar, J. Premalatha, and K. Sathya, "An efficient signcryption scheme for secure authentication using hyper elliptic curve cryptography and keccak hashing," *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 1593–1598, Sep 2019.

Manuscript received 16-11-2021; Accepted 15-5-2022.

■

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 a lecturer at the Faculty of Information 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 - Le Quy Don Technical University. 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 the 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 a lecturer at the Faculty of Information Technology - Le Quy Don Technical University. Research field: Cryptography and Information Security. E-mail: luuhongdung@gmail.com

MỘT DẠNG LƯỢC ĐỒ MÃ HÓA – XÁC THỰC HIỆU NĂNG CAO

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

Tóm tắt

Bài báo đề xuất một dạng lược đồ mã hóa có khả năng xác thực nguồn gốc và tính toàn vẹn của bản tin được mã hóa, mặt khác việc thiết lập khóa bí mật chia sẻ giữa bên gửi/mã hóa và bên nhận/giải mã được thực hiện dựa trên cơ chế của mật mã khóa công khai cho từng bản tin được mã hóa.