Self-Certified Proxy Convertible Authenticated Encryption Scheme

Han-Yu Lin [a,*], Tzong-Sun Wu [b], Ting-Yu Huang [a] and Yi-Shiung Yeh [a]
[a] *Department of Computer Science*
National Chiao Tung University, Taiwan
[b] *Department of Computer Science and Engineering*
National Taiwan Ocean University, Taiwan
* *hanyu.cs94g@nctu.edu.tw*

Abstract

A proxy convertible authenticated encryption (CAE) scheme allows an original signer to delegate his signing power to a proxy signer such that the proxy signer can generate an authenticated ciphertext on behalf of the original signer. The generated authenticated ciphertext can only be decrypted and verified by the specific recipient instead of everyone else for the purpose of confidentiality. Integrating with self-certified public key systems, the proposed scheme can save more communication overheads and computation efforts, since it is not necessary to transmit and verify the public key certificate. That is, authenticating the public key can be combined with subsequent cryptographic operations such as the signature verification. In case of a later repudiation, the specific recipient has the ability to convert the signature into an ordinary one for convincing anyone of the signer's dishonesty.

Keywords: self-certified, proxy signature, convertible, authenticated encryption.

1. Introduction

Since Diffie and Hellman [1] proposed the first public key systems in 1976, public key systems have been widely used in many applications. A critical issue for ensuring the system security is to authenticate the public key before using it. A common used solution is the public key certificate, e.g., X.509 [2]. The certificate is issued by the certification authority and everyone should first verify the corresponding certificate before using the public key. However, some extra communication and computation costs will increase due to the transmission and verification of the certificate. In 1984, Shamir [3] introduced the ID-based public key system in which the public key of each user is straightly his identifier known to the public. Yet, each user's private key is derived by the System Authority (SA) with a trapdoor one-way hash function. That is, the security is entirely relies on the SA and hence a malicious SA can impersonate any legitimate user by deriving his private key without being detected. To overcome the weakness, Girault [4] proposed the so-called self-certified public key system in 1991. A significant property of the self-certified public key system is that the validation of the public key can be combined with the subsequent cryptographic operations such as the signature verification within one step. This will contribute to the reduction of communication and computation costs. In addition, the private key of each user is no longer solely computed by the SA. It can be seen that the self-certified public key system is a better alternative to implement cryptographic schemes as compared with the certificate-based approach or the ID-based system.

In 1994, Horster *et al*. [5] proposed an authenticated encryption (AE) scheme allowing a signer to generate an authenticated ciphertext such that only the designated recipient has the ability to verify it. Such schemes can be applied to many business transactions like the credit card transaction. However, a later dispute on repudiation might occur if the signer denies having generated the signature. To overcome the drawback, Araki *et al*. [6] proposed a convertible limited verifier signature scheme. Yet, their scheme is impracticable, since the signature conversion requires the assistance of the dishonest signer. Moreover, Zhang and Kim [7] also pointed out that Araki *et al*.'s scheme couldn't withstand a universal forgery attack on an arbitrary chosen message. In 2002, Wu and Hsu [8] proposed a convertible authenticated encryption (CAE) scheme, in which the signature conversion is rather simple and can be solely done by the recipient

978-0-7695-3382-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ISDA.2008.139

without any computation efforts or communication overheads. The next year, Huang and Chang [9] also introduced a variant of the CAE schemes. Unfortunately, Lv *et al*. [10] pointed out that the Wu-Hsu scheme and the Huang-Chang scheme could not fulfill the requirement of semantic security.

To meet the need of more and more complicated business developments, Mambo *et al*. [11, 12] proposed the proxy signature schemes for facilitating the delegation operation in an organization. A proxy signature scheme allows the original signer to delegate his signing power to an authorized person called proxy signer, such that the proxy signer can generate a valid proxy signature on behalf of the original one. In this paper, we elaborate on the advantages of self-certified public key systems to propose a novel proxy CAE scheme. The proposed scheme allows a proxy signer to generate an authenticated ciphertext on behalf of the original signer and only the designated recipient has the ability to recover the message and verify its signature for the purpose of confidentiality. In case of a later repudiation, the designated recipient can convert the signature into an ordinary one for the public verification.

The rest of this paper is organized as follows. We present our scheme in Section 2. The security analyses will be discussed in Section 3. Finally, a conclusion is given in Section 4.

2. Self-Certified Proxy CAE Scheme

In this section, we present the proposed scheme over a finite field. Our scheme can be divided into four phases: the user registration, the proxy credential generation, the proxy signature generation and verification, and the proxy signature conversion phases. Initially, the system determines the following public information:

- p, q: two large primes satisfying that $q \mid (p-1)$;
- g: a generator of order q over GF(p);
- $h(\cdot)$: a secure one-way hash function which accepts the input of any length and generates a fixed length output;
- γ: the SA's private key $\gamma \in Z_q^*$;
- β: the SA's public key computed as
$$\beta = g^{\gamma} \bmod p. \quad (1)$$

All the above parameters are made public except for the SA's private key γ. Details of each phase are described as below:

***The user registration phase*:** To join the system, each user U_i associated with the identifier ID_i has to perform the following interactive steps with the SA to obtain his key pair:

Step 1 U_i first chooses an integer $t_i \in Z_q^*$ to compute
$$v_i = g^{h(t_i, ID_i)} \bmod p, \quad (2)$$
and then deliveries (v_i, ID_i) to the SA.

Step 2 Upon receiving (v_i, ID_i), the SA chooses $z_i \in Z_q^*$ to compute
$$y_i = v_i h(ID_i)^{-1} g^{z_i} \bmod p, \quad (3)$$
$$w_i = z_i + h(y_i, ID_i)\gamma \bmod q, \quad (4)$$
and sends (y_i, w_i) back to U_i.

Step 3 U_i computes his private key x_i as
$$x_i = w_i + h(t_i, ID_i) \bmod q, \quad (5)$$
and then ensures its validity by checking
$$\beta^{h(y_i, ID_i)} h(ID_i) y_i \stackrel{?}{=} g^{x_i} \pmod p. \quad (6)$$

If it holds, U_i accepts (x_i, y_i) as his private-and-public key pair. The correctness of Eq. (6) can be easily confirmed as Theorem 1, which also validates the authenticity of y_i with respect to x_i.

Theorem 1. A valid key pair (x_i, y_i) can pass the test of Eq. (6).

Proof: From the left-hand side of Eq. (6), we have

$$\begin{aligned}
\beta^{h(y_i, ID_i)} h(ID_i) y_i &= \beta^{h(y_i, ID_i)} v_i g^{z_i} && \text{(by Eq. (3))} \\
&= v_i g^{z_i + h(y_i, ID_i)\gamma} && \text{(by Eq. (1))} \\
&= g^{h(t_i, ID_i)} g^{z_i + h(y_i, ID_i)\gamma} && \text{(by Eq. (2))} \\
&= g^{h(t_i, ID_i) + w_i} && \text{(by Eq. (4))} \\
&= g^{x_i} \pmod p && \text{(by Eq. (5))}
\end{aligned}$$

which equals to the right-hand side of Eq. (6).

Q.E.D.

***The proxy credential generation phase*:** Let U_o be the original user delegating his signing power to the proxy signer U_p. U_o distributes the proxy credential to U_p with the following steps:

Step 1 U_o first randomly chooses an integer $t \in Z_q^*$ to compute
$$T = g^t \bmod p, \quad (7)$$
$$\sigma = x_o + t(h(m_w, T)) \bmod q, \quad (8)$$
and then sends (σ, m_w, T) to U_p where m_w is the warrant consisting of the identifier of the original and proxy signers, the delegation duration and so on.

Step 2 Upon receiving (σ, m_w, T), U_p verifies

$$g^{\sigma} = \beta^{h(y_o, ID_o)} h(ID_o) y_o T^{h(m_w, T)} \pmod{p} \quad (9)$$

If it holds, U_p proceeds to the next step; else, (σ, m_w, T) is requested to be sent again.

Theorem 2. The verification of Eq. (9) works correctly.

Proof: By raising both sides of Eq. (8) to exponent with base g, we have

$$\begin{aligned} g^{\sigma} &= g^{x_o + th(m_w, T)} \\ &= \beta^{h(y_o, ID_o)} h(ID_o) y_o g^{th(m_w, T)} && \text{(by Eq. (6))} \\ &= \beta^{h(y_o, ID_o)} h(ID_o) y_o T^{h(m_w, T)} \pmod{p} && \text{(by Eq. (7))} \end{aligned}$$

which implies Eq. (9).

Q.E.D.

***The proxy signature generation and verification phase*:** For signing the message m on behalf of the original signer U_o, U_p chooses an integer $k \in Z_q^*$ to compute

$$C = (\beta^{h(y_v, ID_v)} h(ID_v) y_v)^k \bmod p, \quad (10)$$

$$r_1 = m(h(C))^{-1} \bmod p, \quad (11)$$

$$r_2 = h(m, h(g^k \bmod p), C) \bmod q, \quad (12)$$

$$s = k - (x_p + \sigma) h(r_2, T) \bmod q, \quad (13)$$

and then deliveries the proxy signature (m_w, r_1, r_2, s, T) to the designated recipient U_v. Upon receiving the proxy signature (m_w, r_1, r_2, s, T), U_v first computes

$$K = g^s (\beta^{h(y_o, ID_o) + h(y_p, ID_p)} h(ID_o) h(ID_p) \; y_o y_p T^{h(m_w, T)})^{h(r_2, T)} \bmod p, \quad (14)$$

$$C = K^{x_v} \bmod p. \quad (15)$$

He then recovers the message m as

$$m = h(C) r_1 \bmod p, \quad (16)$$

and check the redundancy embedded in m. U_v can further verify the proxy signature (m_w, r_1, r_2, s, T) by checking

$$r_2 = h(m, h(K), C) \bmod q. \quad (17)$$

Theorem 3. With the proxy signature (m_w, r_1, r_2, s, T), the designated recipient U_v can recover the message m and check its validity with Eq. (16).

Proof: From the right-hand side of Eq. (16), we have

$$\begin{aligned} & h(C) r_1 \\ &= h(K^{x_v} \bmod p) r_1 && \text{(by Eq. (15))} \\ &= h((g^s (\beta^{h(y_o, ID_o) + h(y_p, ID_p)} h(ID_o) h(ID_p) \; y_o y_p T^{h(m_w, T)})^{h(r_2, T)})^{x_v} \bmod p) r_1 && \text{(by Eq. (14))} \\ &= h((g^{s + (x_p + \sigma) h(r_2, T)})^{x_v} \bmod p) r_1 && \text{(by Eqs. (9) and (6))} \\ &= h((g^k)^{x_v} \bmod p) r_1 && \text{(by Eq. (13))} \\ &= h((\beta^{h(y_v, ID_v)} h(ID_v) y_v)^k \bmod p) r_1 && \text{(by Eq. (6))} \\ &= h(C) r_1 && \text{(by Eq. (10))} \\ &= m \pmod{p} && \text{(by Eq. (11))} \end{aligned}$$

which leads to the left-hand side of Eq. (16).

Q.E.D.

Theorem 4. If the proxy signature (m_w, r_1, r_2, s, T) is correctly generated, it will pass the test of Eq. (17).

Proof: From the right-hand side of Eq. (17), we have

$$\begin{aligned} & h(m, h(K), C) \\ &= h(m, h(K), K^{x_v} \bmod p) && \text{(by Eq. (15))} \\ &= h(m, h(g^s (\beta^{h(y_o, ID_o) + h(y_p, ID_p)} h(ID_o) h(ID_p) \; y_o y_p T^{h(m_w, T)})^{h(r_2, T)} \bmod p), \\ & \quad (g^s (\beta^{h(y_o, ID_o) + h(y_p, ID_p)} h(ID_o) h(ID_p) \; y_o y_p T^{h(m_w, T)})^{h(r_2, T)} \bmod p)^{x_v} \bmod p) \\ & && \text{(by Eq. (14))} \\ &= h(m, h(g^{s + (x_p + \sigma) h(r_2, T)} \bmod p), \\ & \quad g^{(s + (x_p + \sigma) h(r_2, T)) x_v} \bmod p) && \text{(by Eqs. (9) and (6))} \\ &= h(m, h(g^k \bmod p), g^{k x_v} \bmod p) && \text{(by Eq. (13))} \\ &= h(m, h(g^k \bmod p), C) && \text{(by Eqs. (10) and (6))} \\ &= r_2 \pmod{q} && \text{(by Eq. (12))} \end{aligned}$$

which leads to the left-hand side of Eq. (17).

Q.E.D.

***The proxy signature conversion phase*:** When the case of a later dispute on repudiation occurs, the designated recipient U_v can reveal the converted proxy signature (m_w, r_2, s, C, T) and the original message m to prove the proxy signer's dishonesty without any additional computation efforts or communication overheads. Thus, anyone can verify the converted proxy signature with the assistance of Eqs. (14) and (17).

3. Security Analyses

In this section, we first introduce the definitions of security notions with respect to the proposed scheme, i.e., the discrete logarithm problem (DLP) [1,

13, 14] and the discrete logarithm assumption [13]. We prove that our proposed scheme is secure on condition that the discrete logarithm assumption is intractable.

3.1. Related Definitions

Definition 1 *(discrete logarithm problem; DLP)*

Let (p, q) be two large primes satisfying that $q \mid p-1$ and g a generator of order q over GF(p). The discrete logarithm problem is, given an instance (y, p, q, g) for some $y \in Z_p^*$, to derive $x \in Z_q$ such that $y = g^x \bmod p$. Here, we denote the discrete logarithm $x = \mathrm{Log}_{p,q,g}(y)$.

Definition 2 *(discrete logarithm assumption)*

Let $I_k = \{(p, q, g) \in I \mid |p| = k\}$ with $k \in N$, where I is the universe of all instances and $|p|$ represents the bit-length of p. For every probabilistic polynomial-time algorithm $\mathcal{A}$, every positive polynomial $P(\cdot)$ and all sufficiently large k, the algorithm $\mathcal{A}$ can solve the DLP with an advantage at most $\frac{1}{P(k)}$, i.e.,

$$\Pr[\mathcal{A}(y, p, q, g) = \mathrm{Log}_{p,q,g}(y), (p, q, g) \xleftarrow{u} I_k, y \xleftarrow{u} Z_p^*] \le \frac{1}{P(k)}.$$

Note that "$\xleftarrow{u}$" denotes uniformly and independently selected. The probability is taken over the uniformly and independently chosen instance with a given security parameter k and over the random choices of $\mathcal{A}$.

3.2. Proof of the Proposed Scheme

This subsection proves that the proposed scheme is secure based on the DLP. A problem $\mathcal{P}$ is said to be "(t, ε)-solved" if and only if the problem $\mathcal{P}$ can be solved by a probabilistic polynomial-time (PPT) algorithm $\mathcal{B}$ with the probability ε within polynomial-time t. On the other hand, the PPT algorithm $\mathcal{B}$ is said to "(t, ε)-break" the problem $\mathcal{P}$. We prove that a PPT adversary $\mathcal{A}$ who can (t, ε)-break the DLP is capable of forging a valid proxy signature at most $(t + \tau)$ polynomial-time with the same advantage ε. The detailed proof is given as Theorem 5.

Theorem 5. If the DLP for the instance (y, p, q, g) can be (t, ε)-solved by any PPT adversary $\mathcal{A}$, then he can $(t + \tau, \varepsilon)$-break the proposed scheme where τ is the time required for performing (6 hash function + 4 modular multiplication + 3 modular exponentiation + 1 modular inverse) group operations over GF(p).

Proof:

To forge an authenticated ciphertext on an arbitrarily chosen message m', the adversary $\mathcal{A}$ first obtains a valid converted proxy signature (m_w, r_2, s, C, T) of the message m and computes

$$D = \beta^{h(y_o, ID_o)+h(y_p, ID_p)} h(ID_o) h(ID_p) y_o y_p T^{h(m_w, T)} \bmod p \quad (18)$$

Afterward, the adversary $\mathcal{A}$ randomly and uniformly chooses $a \in Z_q$, and computes

$$W = Dg^a \bmod p. \quad (19)$$

Since a is randomly and uniformly selected from Z_q, we know that W is also uniformly distributed in Z_p^*. It can be seen that the DLP instance (W, p, q, g) for some $W \in Z_p^*$ has the same distribution as any other randomly chosen DLP instances. Solving the DLP for the instance (W, p, q, g) with the advantage ε provides the adversary $\mathcal{A}$ with the value $Z = x_p + \sigma + a$. Consequently, the adversary $\mathcal{A}$ can proceed to compute

$$C' = (\beta^{h(y_v, ID_v)} h(ID_v) y_v)^a \bmod p, \quad (20)$$

$$r_1' = m' h(C')^{-1} \bmod p, \quad (21)$$

$$r_2' = h(m', h(g^a \bmod p), C') \bmod q, \quad (22)$$

$$s' = a - (Z - a) h(r_2', T) \bmod q. \quad (23)$$

Here, (m_w, r_1', r_2', s', T) is the forged authenticated ciphertext of the message m'. Since the time required for computing Z is at most t and that, denoted by τ, for computing (r_1', r_2', s') is (6 hash function + 4 modular multiplication + 3 modular exponentiation + 1 modular inverse) group operations over GF(p), we conclude that the total execution time is bounded by $t + \tau$, which is also polynomial-time. The validity of the forged proxy signature (m_w, r_1', r_2', s', T) can be verified as follows:

From the right-hand side of Eq. (17), we have

$$\begin{aligned} & h(m', h(K'), C') \\ &= h(m', h(g^{s'} (\beta^{h(y_o, ID_o)+h(y_p, ID_p)} h(ID_o) h(ID_p) y_o y_p T^{h(m_w, T)})^{h(r_2, T)} \bmod p), C') \text{ (by Eq. (14))} \\ &= h(m', h(g^{s'+(x_p+\sigma)h(r_2', T)} \bmod p), C') \text{ (by Eqs. (9) and (6))} \\ &= h(m', h(g^{s'+(Z-a)h(r_2', T)} \bmod p), C') \\ &= h(m', h(g^a \bmod p), C') \text{ (by Eq. (23))} \\ &= r_2' \pmod q \text{ (by Eq. (22))} \end{aligned}$$

which leads to the left-hand side of Eq. (17). It can be

seen that the forged authenticated ciphertext (m_w, r_1', r_2', s', T) will successfully pass the test of Eq. (17).

Q.E.D.

4. Conclusions

In this paper, we have proposed a self-certified proxy convertible authenticated encryption (CAE) scheme which allows the proxy signer to generate a valid authenticated ciphertext on behalf of the original signer, such that only the designated recipient has the ability to recover the message and verify its corresponding signature. One significant characteristic of our scheme is that validating the public key and verifying the signature can be simultaneously carried out within one step, which helps reducing the communication overheads and computation efforts. In case of a later repudiation, the designated recipient has the ability to solely release the converted proxy signature for the public verification. Moreover, we also proved that the proposed scheme is secure on condition that the discrete logarithm assumption is intractable.

5. References

[1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT-22, No. 6, 1976, pp. 644-654.

[2] ISO/IEC 9594-8, "Information technology – open systems interconnection – the directory: public-key and attribute certificate frameworks," International Organization for Standardization, 2001.

[3] A. Shamir, "Identity-based cryptosystems and signature schemes," *Advances in Cryptology – CRYPTO'84*, Springer-Verlag, 1984, pp. 47-53.

[4] M. Girault, "Self-certified public keys," *Advances in Cryptology – EUROCRYPT'91*, Springer-Verlag, 1991, pp. 491-497.

[5] P. Horster, M. Michel and H. Peterson, "Authenticated encryption schemes with low communication costs," *Electronics letters*, Vol. 30, No. 15, 1994, pp. 1212-1213.

[6] S. Araki, S. Uehara and K. Imamura, "The limited verifier signature and its application," *IEICE Transactions on Fundamentals*, Vol. E82-A, No. 1, 1999, pp. 63-68.

[7] F. Zhang and K. Kim, "A universal forgery on Araki *et al.*'s convertible limited verifier signature scheme," *IEICE Transactions on Fundamentals*, Vol. E86-A, No. 2, 2003, pp. 515-516.

[8] T.S. Wu and C.L. Hsu, "Convertible authenticated encryption scheme," *The Journal of Systems and Software*, Vol. 62, No. 3, 2002, pp. 205-209.

[9] H. Huang, C. Chang, "An efficient convertible authenticated encryption scheme and its variant," *Proceedings of the ICICS2003-Fifth International Conference on Information and Communications Security*, LNCS 2836, Springer-Verlag, Berlin, 2003, pp. 382-392.

[10] J. Lv, X. Wang and K. Kim, "Practical convertible authenticated encryption schemes using self-certified public keys," *Applied Mathematics and Computation*, Vol. 169, No. 2, 2005, pp. 1285-1297.

[11] M. Mambo, K. Usuda and E. Okamoto, "Proxy signature for delegating signature operation," *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, ACM press, 1996, pp. 48-57.

[12] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures: delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronic Communications and Computer Science*, Vol. E79-A, No. 9, 1996, pp. 1338-1354.

[13] H. Delfs and H. Knebl, *Introduction to Cryptography: Principles and Applications*, Springer, 2002.

[14] A. Menezes, P. Oorschot and S. Vanstone, *Handbook of applied cryptography*, CRC Press, Inc., 1997.