Copyright ⓒ 2014 KSII

Toward Efficient Convertible Authenticated Encryption Schemes Using Self-Certified Public Key System

Tzong-Sun Wu*, Yih-Sen Chen and Han-Yu Lin
Department of Computer Science and Engineering, National Taiwan Ocean University
Keelung, 202, Taiwan
[e-mail: ibox456@gmail.com]
[e-mail: ys3889@gmail.com]
[e-mail: lin.hanyu@msa.hinet.net]
*Corresponding author: Tzong-Sun Wu

Received November 15, 2013; revised February 12, 2014; accepted March 5, 2014; published March 31, 2014

Abstract

Convertible authenticated encryption (CAE) schemes enable the signer to send a confidential message and its corresponding signature to the designated recipient. The recipient can also convert the signature into a conventional one which can be verified by anyone. Integrating the properties of self-certified public key systems, this paper presents efficient and computationally indistinguishable self-certified CAE schemes for strengthening the security of E-Commerce applications. Additionally, we also adapt the proposed schemes to elliptic curve systems for facilitating the applications of limited computing power and insufficient storage space. The proposed schemes are secure against known existential active attacks, satisfy the semantic security requirement, and have the following advantages: (i) No extra certificate is required since the tasks of authenticating the public key and verifying the signature can be simultaneously carried out within one step, which helps reducing computation efforts and communication overheads. (ii) In case of a later dispute, the recipient can convert the signature into an ordinary one for the public arbitration. (iii) The signature conversion can be solely performed by the recipient without additional computation efforts or communication overheads. (iv) The recipient of the signature can prove himself, if needed, to anyone that he is actually the designated recipient.

Keywords: authenticated encryption, self-certified public key, convertibility, elliptic curve, semantic security

http://dx.doi.org/10.3837/tiis.2014.03.026

1. Introduction

In the field of cryptography, how to satisfy the requirements of integrity [1], confidentiality [2], authenticity [1] and non-repudiation [3] over open environments of the Internet is always an important issue. In 1976, Diffie and Hellman [4] introduced the first public key system based on the intractability of the discrete logarithm problem (DLP) [4, 5]. In the system, each user owns a self-chosen private key and a corresponding public key stored in the public key directory. One can use his private key along with the designated user's public key to compute a shared common key and thus to construct a secure channel between them. With the public key cryptosystems [6, 7], we can further perform the functions of the public key encryption or the digital signature. However, a malicious adversary can plot an active attack by substituting a fake public key for the genuine one.

To withstand the above attack, a certificate-based approach (e.g., X.509) [8] is a commonly used solution. Each public key is accompanied with a certificate issued by the certification authority to guarantee its authenticity. One should first verify the public key before using it. However, it requires extra communication overheads and computation efforts owing to the processes of transmitting and verifying the certificate.

In 1984, Shamir [9] addressed the concept of ID-based public key systems in which each user's public key is straightly his public identifier, so as to be explicitly verified without any extra certificate. The corresponding private key is derived by the system authority (SA) through a trapdoor one-way hash function which is computationally infeasible to invert. Without the SA's secret information, no one can obtain a valid private key. Nevertheless, the SA can still impersonate any legitimate user without being detected since he has the control over every user's private key. That is, the SA should be always trusted.

Seeing that the security of ID-based public key systems places great dependence on the SA and users cannot freely choose his own private key, Girault [10] proposed a self-certified public key system to eliminate these drawbacks in 1991. A self-certified public key system has the property that the public key validation can be combined with other subsequent cryptographic mechanisms such as the signature verification. That is, the tasks of authenticating the public key and verifying the signature can be simultaneously achieved in one step, which reduces the costs of the certificate transmission and public key verification. As compared with the stated two systems, a self-certified public key system is more efficient. It might be a better alternative for implementing cryptographic systems.

To meet requirements of some specific applications that digital signatures must simultaneously fulfill the need of confidentiality, such as the delivery of military documents or transactions of credit cards, a flat-out way would be the conventional two-step approach [11], i.e., first sign then encrypt. However, the two-step approach is inefficient since the costs equal to the sum of those of signing and encryption. To improve the efficiency, an authenticated encryption scheme was proposed by Horster *et al*. [12] in 1994, which only allows a designated recipient to verify the signature rather than anyone else for the purpose of confidentiality. Obviously, the authenticated encryption scheme outperforms the traditional two-step approach in terms of computation complexities and communication overheads. In 2005, Yoon and Yoo [13] extended the applications for message linkages. Yet, a later dispute that the signer disclaims having generated a signature might occur. To convince anyone of the signer's dishonesty, the designated recipient must have the ability to

Copyright ⓒ 2014 KSII

convert the signature into an ordinary one for protecting his rights or benefits. In 1999, Araki *et al*. [14] put the concept of signature conversion into practice and proposed a convertible limited verifier signature scheme. However, the process of signature conversion requires the assistance of the signer, which might be infeasible if the signer is unwilling to cooperate with. To improve the conversion mechanism and obtain better performance, Wu and Hsu [15] proposed a convertible authenticated encryption (CAE) scheme in which the signature conversion process is rather simple and can be solely done by the recipient without any computation efforts. Their scheme has to perform extra public key verification before any cryptographic mechanism and does not provide the property of recipient proof. Chen and Jan [16] further proposed an enhanced scheme in the same year. However, both the Wu-Hsu and the Chen-Jan schemes cannot provide the semantic security, i.e., the ciphertext is computationally distinguishable with respect to two candidate messages. To eliminate such security weakness, Lv *et al.* [17] proposed a secure and practical solution. Yet, the computation complexity of their scheme is rather high. Since then, many variations of self-certified CAE were proposed, for instance, self-certified proxy CAE [18, 19] and convertible multi-authenticated encryption scheme [20, 21], which can be used in different application situations. Interested readers can refer to these literatures [15, 17-21] for more detailed discussions. Elaborating on the merits inherent in the self-certified public key systems, the authors will propose efficient and computationally indistinguishable self-certified CAE schemes in this paper.

With the coming of the digitalized time, lots of mobile devices like mobile phones have been widely used around. Equipped with less powerful computing capability and small storage space, those devices can only be used to execute fewer computations and store limited personal sensitive data. No one can gain the access to the data inside without authorization. Due to the limited computing power and storage space, the computation complexity and the storage requirement are concerned the most when we implement a cryptographic scheme for those devices. The elliptic curve cryptography (ECC) [22-27] first introduced by Koblitz [28] and Miller [29] is applicable to this kind of applications used in mobile devices. A significant characteristic of the ECC is that the key length is shorter than that of the conventional cryptography under the same level of security, which helps faster execution and more bandwidth savings. As we elevate the security level, the difference of the key length between the ECC and the conventional cryptography dramatically increases. Therefore, we also adapt the proposed schemes to elliptic curve systems for facilitating the gradually widely used applications of limited computing power and insufficient storage. To ensure the authenticity of transaction certificates is the foundation for any secure E-commerce application. In our proposed schemes, the tasks of authenticating the public key, corresponding certificates and the signature can be simultaneously carried out in one logical step, which greatly reduces the costs of transmitting certificates and verifying public keys. Moreover, under the same security level, the elliptic curve variants with shorter key length helps with faster execution and more bandwidth savings. Therefore, our proposed schemes can strengthen the security of E-commerce application.

2. Self-Certified CAE Schemes Based on Discrete Logarithms

The proposed self-certified CAE schemes are divided into four stages: the user registration, the signature generation and verification, the signature conversion, and the recipient proof stages. There is also a system authority (SA) whose tasks are to initialize the system and to

help users generating their key pairs. Initially, the SA chooses the following necessary parameters:

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 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

$$\beta = g^{\gamma} \bmod p. \tag{1}$$

All the above parameters are made public except for the SA's private key γ. Details of each stage are described as below and shown in **Fig. 1**:

***The user registration stage*:** 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 a private-public key pair:

Step 1 U_i first chooses an integer $t_i \in Z_q^*$ to compute

$$v_i = g^{h(t_i, ID_i)} \bmod p, \tag{2}$$

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

$$y_i = v_i h(ID_i)^{-1} g^{z_i} \bmod p, \tag{3}$$
$$w_i = z_i + h(y_i, ID_i)\gamma \bmod q, \tag{4}$$

and sends (y_i, w_i) back to U_i.

Step 3 U_i computes his private key x_i as

$$x_i = w_i + h(t_i, ID_i) \bmod q, \tag{5}$$

and then ensures its validity by checking

$$\beta^{h(y_i, ID_i)} h(ID_i) y_i \stackrel{?}{=} g^{x_i} \pmod p. \tag{6}$$

If it holds, U_i accepts (x_i, y_i) as his private-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

$$\beta^{h(y_i, ID_i)} h(ID_i) y_i = \beta^{h(y_i, ID_i)} v_i g^{z_i} \quad \text{(by Eq. (3))}$$

Copyright ⓒ 2014 KSII

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

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

Q.E.D.

***The signature generation and verification stage*:** When U_a wants to send U_b the signature of a confidential message m with embedded redundancy. U_a first chooses an integer $k \in z_q^*$ to compute (C, r_1, r_2) as Eqs. (7), (8) and (9), respectively:

$$C = (\beta^{h(y_b, ID_b)} h(ID_b) y_b)^k \bmod p, \tag{7}$$
$$r_1 = mh(C)^{-1} \bmod p, \tag{8}$$
$$r_2 = h(m, h(g^k \bmod p), C) \bmod q. \tag{9}$$

U_a then compute s as Eq. (10.*) in **Table 1**, where '*' represents one letter of 'a' to 'f'. Each equation is a secure combination of three parameters k, x_a and r_2.

Table 1. Equations to generate signature s

I	$s = k(1 + x_a r_2)^{-1} \bmod q$	(10.a)
II	$s = x_a^{-1}(kr_2 - 1) \bmod q$	(10.b)
III	$s = k - x_a r_2 \bmod q$	(10.c)
IV	$s = k^{-1}(r_2 + x_a) \bmod q$	(10.d)
V	$s = kr_2 + x_a \bmod q$	(10.e)
VI	$s = x_a^{-1}(k - r_2) \bmod q$	(10.f)

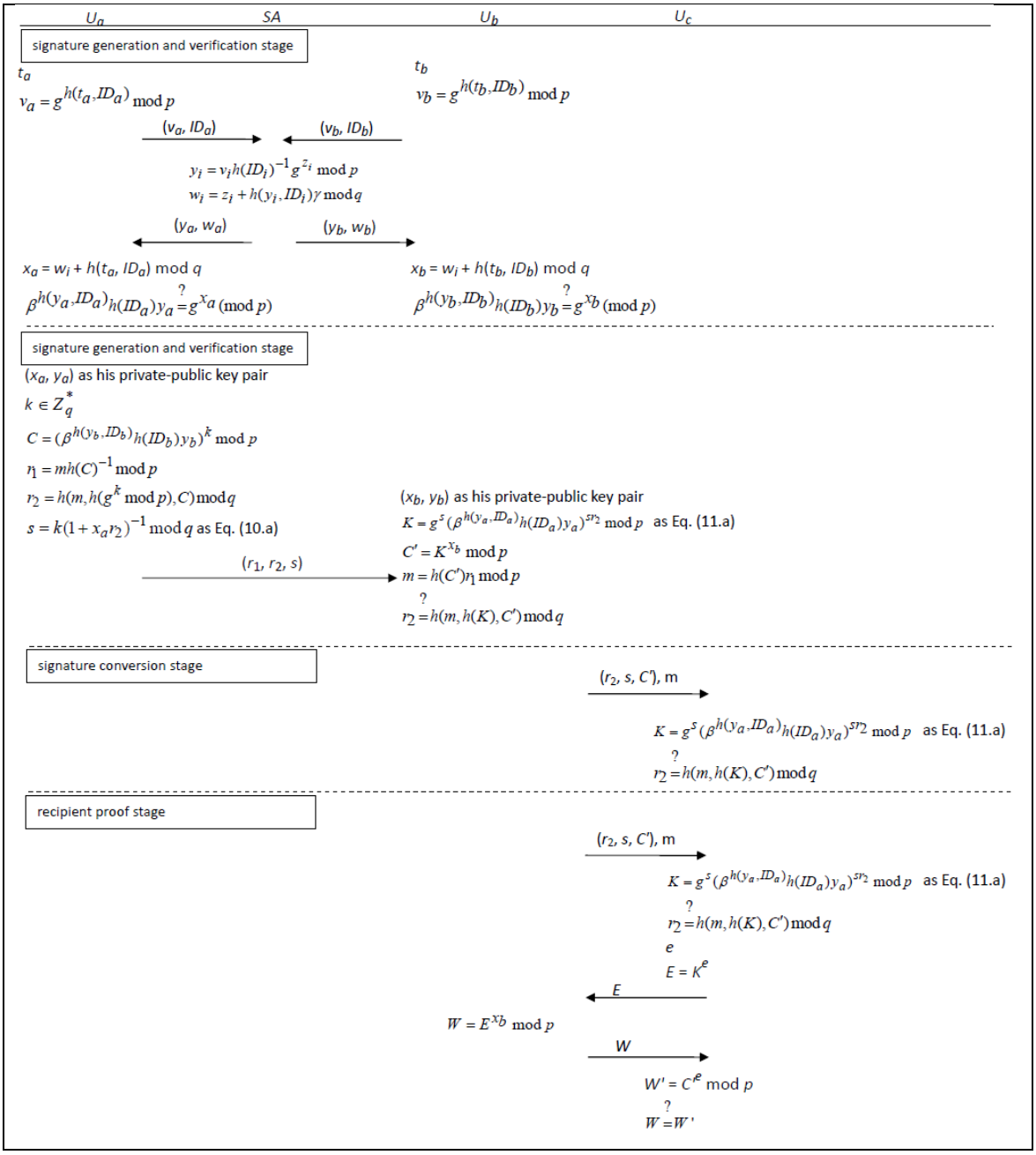

Fig. 1. Illustration of proposed self-certified CAE schemes based on Discrete Logarithms

Copyright ⓒ 2014 KSII

Table 2. Equations to compute K

I	$K = g^{s}(\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{sr2} \bmod p \frac{1}{2}$	(11.a)
II	$K = (g(\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{s})^{r_2^{-1}} \bmod p$	(11.b)
III	$K = g^{s}(\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{r2} \bmod p$	(11.c)
IV	$K = (g^{r2}(\beta^{h(y_a, ID_a)} h(ID_a) y_a))^{s^{-1}} \bmod p$	(11.d)
V	$K = (g^{s}(\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{-1})^{r_2^{-1}} \bmod p$	(11.e)
VI	$K = g^{r2}(\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{s} \bmod p$	(11.f)

Here, (r_1, r_2, s) is the signature for m, which is then delivered to the designated recipient U_b. After receiving the signature, U_b first computes K from Eq. (11.*) of **Table 2** and C' from Eq. (12). Note that the computation of K depends on the generation of s, e.g., generating s with Eq. (10.c) implies deriving K with Eq. (11.c).

$$C' = K^{xb} \bmod p \tag{12}$$

The message m with the embedded redundancy can be recovered by Eq. (13).

$$m = h(C')r_1 \bmod p \tag{13}$$

U_b finally verifies signature (r_1, r_2, s) by checking the following equation:

$$r_2 = h(m, h(K), C') \bmod q \tag{14}$$

If it holds, the signature is valid. In the mean time, the signer's public key y_a is simultaneously authenticated.

Take scheme I (with s and K separately computed as Eqs. (10.a) and (11.a)) as an example. We demonstrate that Eqs. (13) and (14) work correctly as the proofs of Theorems 2 and 3, respectively. The correctness of the other schemes can be assured with the similar way.

Theorem 2. The designated recipient U_b can recover the message m with Eq. (13).
Proof: From the right-hand side of Eq. (13), we have

$$\begin{aligned}
& h(C')r_1 \\
&= h(K^{xb} \bmod p)r_1 && \text{(by Eq. (12))} \\
&= h(((\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{sr2} g^{s})^{xb} \bmod p)r_1 && \text{(by Eq. (11.a))} \\
&= h((g^{s(1+x_a r_2)})^{xb} \bmod p)r_1 && \text{(by Eq. (6))} \\
&= h(g^{x_b k} \bmod p)r_1 && \text{(by Eq. (10.a))}
\end{aligned}$$

$$= h((\beta^{h(y_b, ID_b)} {}_{h(ID_b)} y_b)^k \bmod p) r_1 \quad \text{(by Eq. (6))}$$
$$= h(C) r_1 \quad \text{(by Eq. (7))}$$
$$= m \pmod p \quad \text{(by Eq. (8))}$$

which equals to the left-hand side of Eq. (13).

Q.E.D.

Theorem 3. The tasks of verifying the signature (r_1, r_2, s) and authenticating the public key y_a can be simultaneously achieved with Eq. (14).

Proof: From the right-hand side of Eq. (14), we have

$$h(m, h(K), C')$$
$$= h(m, h(K), K^{x_b} \bmod p) \quad \text{(by Eq. (12))}$$
$$= h(m, h(g^s (\beta^{h(y_a, ID_a)} {}_{h(ID_a)} y_a)^{sr_2} \bmod p),$$
$$(g^s (\beta^{h(y_a, ID_a)} {}_{h(ID_a)} y_a)^{sr_2} \bmod p)^{x_b} \bmod p) \quad \text{(by Eq. (11.a))}$$
$$= h(m, h(g^{s(1+x_a r_2)} \bmod p), g^{s(1+x_a r_2) x_b} \bmod p)$$
$$= h(m, h(g^k \bmod p), g^{k x_b} \bmod p) \quad \text{(by Eq. (10.a))}$$
$$= h(m, h(g^k \bmod p), (\beta^{h(y_b, ID_b)} {}_{h(ID_b)} y_b)^k \bmod p) \quad \text{(by Eq. (6))}$$
$$= h(m, h(g^k \bmod p), C) \quad \text{(by Eq. (7))}$$
$$= r_2 \pmod q \quad \text{(by Eq. (9))}$$

which equals to the left-hand side of Eq. (14).

Q.E.D.

The signature conversion stage: When a later dispute of signer's repudiation occurs, the recipient U_b can show the dishonesty of the signer by releasing the converted signature (r_2, s, C') along with the recovered message m. Anyone can first compute K from Eq. (11.*) and then validate the signature with Eq. (14). If the checking of Eq. (14) holds, he assures that the signature is generated by U_a.

***The recipient proof stage*:** For convincing someone, say, U_c, that he is the real recipient, the recipient U_b can perform the following interactive steps with U_c:

Step 1 U_b sends (r_2, s, C') and m to U_c.

Step 2 U_c first computes K with the corresponding Eq. (11.*) and then checks the signature's validity with Eq. (14). If it holds, U_c proceeds to the next step; otherwise, the protocol is terminated.

Step 3 U_c randomly chooses an integer e to compute $E = K^e \bmod p$ and then transmits E to U_b.

Step 4 Upon receiving E, U_b computes $W = E^{x_b} \bmod p$ and returns it to U_c.

Step 5 U_c computes $W' = C'^{\,e} \bmod p$ and checks whether $W = W'$. If it holds, U_c is convinced that U_b is the designated recipient.

Copyright ⓒ 2014 KSII

3. Self-Certified CAE Schemes Based on Elliptic Curve Discrete Logarithms

In this section, we present elliptic curve variants of proposed schemes based on the elliptic curve discrete logarithm problem (ECDLP) [8, 13, 14]. The stages and the participating parties of the proposed elliptic curve variants are the same as those in the proposed schemes in Section 2. Initially, the SA determines the following parameters:

p: a large prime;

a, b: two parameters in Z_p satisfying that $4a^3 + 27b^2 \bmod p \neq 0$;

$E_p(a, b)$: an elliptic curve over GF(p) containing a set of points (x, y) satisfying that $y^2 = x^3 + ax + b \pmod p$;

O: a point at infinity over $E_p(a, b)$;

G: the base point of order q over $E_p(a, b)$, where q is a large prime;

$h(\cdot)$: a secure one-way hash function which accepts input of various length and generates output of a fixed length; note that the input of a point over $E_p(a, b)$ represents the input of the concatenation of the x- and y- coordinates of that point;

γ: the SA's private key for $\gamma \in Z_q^*$;

B: the SA's public key computed as

$$B = \gamma G \text{ over } E_p(a, b). \tag{15}$$

All of the above parameters are made public except for the SA's private key γ. In the following, all elliptic curve point operations are manipulated over $E_p(a, b)$. Details of each stage are shown as follows:

***The user registration stage*:** To become a legitimate user, U_i associated with the identifier ID_i performs the registration process with the SA.

Step 1 U_i first chooses an integer $t_i \in Z_q^*$ to compute

$$V_i = h(t_i, ID_i)G, \tag{16}$$

and then sends (V_i, ID_i) to the SA.

Step 2 After receiving (V_i, ID_i), the SA chooses $z_i \in Z_q^*$ to compute

$$Y_i = (h(ID_i)^{-1} \bmod q)(V_i + z_i G), \tag{17}$$

$$w_i = z_i + h(Y_i, ID_i)\gamma \bmod q, \tag{18}$$

and returns (Y_i, w_i) to U_i.

Step 3 U_i computes x_i as

$$x_i = w_i + h(t_i, ID_i) \bmod q, \tag{19}$$

and checks its validity with the following equality.

$$h(Y_i, ID_i)B + h(ID_i)Y_i \stackrel{?}{=} x_i G \,. \tag{20}$$

If the above equation holds, U_i accepts (x_i, Y_i) as his private and public keys. Theorem 4 proves the correctness of Eq. (20) which also validates the authenticity of Y_i with respect to x_i.

Theorem 4. U_i can perform Eq. (20) to authenticate the public key Y_i with respect to his private key x_i.

Proof: From the left-hand side of Eq. (20), we have

$$\begin{aligned} h(Y_i, ID_i)B + h(ID_i)Y_i &= h(Y_i, ID_i)B + (V_i + z_i G) && \text{(by Eq. (17))} \\ &= (h(Y_i, ID_i)\gamma + z_i)G + V_i && \text{(by Eq. (15))} \\ &= w_i G + V_i && \text{(by Eq. (18))} \\ &= (w_i + h(t_i, ID_i))G && \text{(by Eq. (16))} \\ &= x_i G && \text{(by Eq. (19))} \end{aligned}$$

which equals to the right-hand side of Eq. (20).

Q.E.D.

***The signature generation and verification stage*:** When U_a wants to send U_b the signature of a confidential message m with embedded redundancy. U_a first chooses an integer $k \in Z_q^*$ and computes (C, r_1, r_2, s) as Eqs. (21), (8), (12) and (10.*) of **Table 1**, respectively.

$$C = k(h(Y_b, ID_b)B + h(ID_b)Y_b) \tag{21}$$

$$r_2 = h(m, h(kG), C) \bmod q \tag{22}$$

(r_1, r_2, s) is the signature for m, which is then sent to the designated recipient U_b. Upon receiving the signature, U_b first computers the corresponding K from Eq. (23.*) of **Table 3** and C' from Eq. (24).

$$C' = x_b K \tag{24}$$

The message m with the embedded redundancy can be recovered from Eq. (13). After that, U_b can verify the signature (r_1, r_2, s) by testing Eq. (14). If it holds, the signature is valid; meanwhile, the signer's public key y_a is simultaneously authenticated.

Taking scheme I (with s and K separately computed as Eqs. (10.a) and Eq. (23.a)) as an example, we demonstrate that the proposed elliptic variants work correctly as the proofs of Theorems 5 and 6, respectively. Interested readers can follow the similar way to check the correctness of the other schemes.

Theorem 5. The recipient U_b can recover the message m with Eq. (13).

Proof: From the right-hand side of Eq. (13), we have

Copyright ⓒ 2014 KSII

$$
\begin{aligned}
& h(C')r_1 \\
&= h(x_b K)r_1 && \text{(by Eq. (24))} \\
&= h(sx_b(r_2(h(Y_a, ID_a)B + h(ID_a)Y_a) + G))r_1 && \text{(by Eq. (23.a))} \\
&= h((sx_b)(r_2 x_a G + G))r_1 && \text{(by Eq. (20))} \\
&= h(sx_a r_2 x_b G + sx_b G)r_1 \\
&= h((sx_a r_2 + s)(x_b G))r_1 \\
&= h((sx_a r_2 + s)(h(Y_b, ID_b)B + h(ID_b)Y_b))r_1 && \text{(by Eq. (20))} \\
&= h(k(h(Y_b, ID_b)B + h(ID_b)Y_b))r_1 && \text{(by Eq. (10.a))} \\
&= h(C)r_1 && \text{(by Eq. (21))} \\
&= m \pmod p && \text{(by Eq. (8))}
\end{aligned}
$$

which equals to the left-hand side of Eq. (13).

Q.E.D.

Table 3. Equations to compute K based on the ECDLP

I	$K = s(r_2(h(Y_a, ID_a)B + h(ID_a)Y_a) + G)$	(23.a)
II	$K = r_2^{-1}(s(h(Y_a, ID_a)B + h(ID_a)Y_a) + G)$	(23.b)
III	$K = r_2(h(Y_a, ID_a)B + h(ID_a)Y_a) + sG$	(23.c)
IV	$K = s^{-1}(h(Y_a, ID_a)B + h(ID_a)Y_a + r_2 G)$	(23.d)
V	$K = r_2^{-1}(-(h(Y_a, ID_a)B + h(ID_a)Y_a) + sG)$	(23.e)
VI	$K = s(h(Y_a, ID_a)B + h(ID_a)Y_a) + r_2 G$	(23.f)

Theorem 6. A valid signature (r_1, r_2, s) should satisfy Eq. (14) which also authenticates the public key Y_a.

Proof: From the right-hand side of Eq. (14), we have

$$
\begin{aligned}
& h(m, h(K), C') \\
&= h(m, h(K), x_b K) && \text{(by Eq. (24))} \\
&= h(m, h(sG + sr_2(h(Y_a, ID_a)B + h(ID_a)Y_a)), \\
&\quad x_b(sG + sr_2(h(Y_a, ID_a)B + h(ID_a)Y_a))) && \text{(by Eq. (23.a))} \\
&= h(m, h((sr_2)x_a G + sG), x_b((sr_2)x_a G + sG)) && \text{(by Eq. (20))} \\
&= h(m, h((sx_a r_2 + s)G), x_b(sx_a r_2 + s)G) \\
&= h(m, h(kG), x_b kG) && \text{(by Eq. (10.a))} \\
&= h(m, h(kG), k(h(Y_b, ID_b)B + h(ID_b)Y_b)) && \text{(by Eq. (20))} \\
&= h(m, h(kG), C) && \text{(by Eq. (21))} \\
&= r_2 \pmod q && \text{(by Eq. (22))}
\end{aligned}
$$

which equals to the left-hand side of Eq. (14).

Q.E.D.

The signature conversion stage: To handle the case of a later dispute, the recipient U_b can simply reveal the recovered message m and the converted signature (r_2, s, C'). Anyone can first compute K from Eq. (23.*) and then validate the signature with Eq. (14). If the checking of Eq. (14) holds, he assures that the signature is generated by U_a.

***The recipient proof stage*:** For convincing someone, say, U_c, that he is the real recipient, the recipient U_b can perform the following interactive steps with U_c:

Step 1 U_b sends (r_2, s, C') to U_c.

Step 2 U_c first computes K with corresponding Eq. (23.*) and then checks the signature's validity with Eq. (14). If it holds, U_c proceeds to the next step; otherwise, the protocol is terminated.

Step 3 U_c randomly chooses an integer e to compute $E = eK$ and then transmits E to U_b.

Step 4 Upon receiving E, U_b computes $W = x_bE$ and returns it to U_c.

Step 5 U_c computes $W' = eC'$ and checks whether $W = W'$. If it holds, U_c is convinced that U_b is the designated recipient.

4. Security Considerations and Performance Evaluation

In this section, we first defined some security notions for self-certified CAE schemes and then gave security proofs and the performance evaluation of our proposed schemes.

4.1 Security Notions

To facilitate the following proofs, we regenerate algorithms of User-registration, Signature-encryption-and-verification, Signature-conversion and Recipient-proof from each phase of the proposed schemes. The security notions of message confidentiality and unforgeability with respect to self-certified CAE schemes are defined below.

Message Confidentiality. A self-certified CAE scheme can fulfill the security requirement of message confidentiality if authenticated ciphertexts are indistinguishable under chosen ciphertext attacks. We define a security model for indistinguishability of authenticated ciphertexts under chosen ciphertext attacks. In this model, the adversary attempts to decrypt a target ciphertext of the designated recipient.

Definition 1. *A self-certified CAE scheme is said to be semantically secure against chosen ciphertext attacks if there exits no polynomial-time adversary with a non-negligible advantage in the following game*:

Setup: A challenger $\mathcal{C}$ first generates necessary system parameters and then obtains the signer U_a's key pair (x_a, y_a) by the User-registration algorithm. System parameters and the signer U_a's public key are given to an adversary $\mathcal{A}$. Upon receiving these parameters, the adversary $\mathcal{A}$ determines one designated recipient U_b*. The recipient U_b*'s public key y_b* can be acquired from the User-registration algorithm, but the corresponding private key x_b* is unknown to $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ can issue several kinds of queries adaptively:

– Signature-encryption-and-verification queries: The adversary $\mathcal{A}$ can query either

Copyright ⓒ 2014 KSII

Signature-encryption or Signature-verification. In the Signature-encryption queries, the adversary $\mathcal{A}$ produces a message m with respect to U_a and sends it to the challenger $\mathcal{C}$ which then returns the result of Signature-encryption (m, x_a, y_b*) to $\mathcal{A}$. In the Signature-verification queries, the adversary $\mathcal{A}$ produces an authenticated ciphertext σ = (r_1, r_2, s) and requests the result of Signature-verification (σ, y_a, x_b*) with respect to U_a and U_b* from the challenger $\mathcal{C}$. If the recovered message is consistent with the redundancy check and its corresponding signature is valid, $\mathcal{C}$ responses the message; Otherwise, the $\perp$ symbol is returned as a result.

– Signature-conversion queries: The adversary $\mathcal{A}$ produces an authenticated ciphertext σ = (r_1, r_2, s) and requests the result of Signature-conversion (σ, y_a, x_b*) with respect to U_a and U_b* from the challenger $\mathcal{C}$. If the result (r_2, s, C') is a valid converted signature for the message m with suitable redundancy, $\mathcal{C}$ responses the result; Otherwise, the $\perp$ symbol is returned as a result.
– Recipient-proof queries: The adversary $\mathcal{A}$ produces an authenticated ciphertext σ = (r_1, r_2, s) and requests the result of Recipient-proof (σ, y_a, x_b*) with respect to U_a from the challenger $\mathcal{C}$. If U_b* is the designated recipient, $\mathcal{C}$ responses the symbol 1; Otherwise, the $\perp$ symbol is returned as a result.

Challenge: The adversary $\mathcal{A}$ produces two messages, m_0 and m_1, of the same length. The challenger $\mathcal{C}$ flips a coin $\lambda \leftarrow \{0, 1\}$ and generates an authenticated ciphertext σ* = Signature-encryption (m_λ, x_a, y_b*) which is then delivered to $\mathcal{A}$ as a target challenge.

Phase 2: The adversary $\mathcal{A}$ can issue new queries as those in Phase 1, except that the Signature-verification or Signature-conversion query for the target challenge σ* is prohibited.

Guess: At the end of the game, $\mathcal{A}$ outputs a bit λ'. The adversary $\mathcal{A}$ wins this game if $\lambda' = \lambda$. We define $\mathcal{A}$'s advantage as $Adv(\mathcal{A}) = \Pr[\lambda' = \lambda] - 1/2$.

Unforgeability. A cryptographic scheme satisfies the security requirement of unforgeability if it is secure against chosen message attacks. We define a model for unforgeability of self-certified CAE scheme against chosen message attacks. In this model, the adversary attempts to forge a valid signature of one target message.

Definition 2. *A self-certified CAE scheme is said to achieve existential unforgeability against chosen-message attacks if there exists no polynomial-time adversary with a non-negligible advantage in the following game:*

Setup: A challenger $\mathcal{C}$ first generates necessary system parameters, and then obtains the signer U_a's key pair (x_a, y_a) and a designated recipient U_b's key pair (x_b, y_b) by the User-registration algorithm. The challenger $\mathcal{C}$ then gives the forger $\mathcal{F}$ system parameters, the signer U_a's public key y_a and the designated recipient's public key y_b.

Attack: The forger $\mathcal{F}$ issues the same queries as those in Phase 1 of Definition 1.

Forgery: Finally, $\mathcal{F}$ produces an authenticated ciphertext σ*. The forger $\mathcal{F}$ wins if σ* can be converted into a valid signature (r_2*, s*, C'*) for some message m* with redundancy by the designated recipient. Note that it is not allowed to issue a Signature-encryption query for m*.

4.2 Security Proof

The security of the proposed schemes is based on the DLP [4, 5] / ECDLP [25, 30, 31] and security of Nyberg-Rueppel signature schemes [32, 33]. For the details of Nyberg-Rueppel

signature schemes, we recommend the interested readers to refer to [32, 33]. Instead of separate discussions, we only take the DLP-based scheme I as an instance for the following proofs. Other schemes can be proved with similar ways. The definition of DLP is briefly restated below: Let p be a large prime, g a generator, and α a random integer. It is computationally infeasible to derive α from known (g, $g^{\alpha} \bmod p$). In the following, we proved that the proposed schemes satisfy the security requirements of confidentiality and unforgeability as Theorems 7 and 8, respectively.

Theorem 7. *The proposed self-certified CAE scheme is* (t, ε)*-secure against chosen ciphertext attacks if there exists no polynomial-time algorithm* β_1 *that can* (t_1, ε_1)*-break the DLP.*

Proof. Suppose that $\mathcal{A}$ is a (t, ε)-algorithm that breaks the self-certified CAE scheme under the chosen ciphertext attack, where t denotes the running time and ε the probability that $\mathcal{A}$ succeeds. We will show that we can use $\mathcal{A}$ to construct a (t_1, ε_1)-algorithm β_1 that solves the DLP in time t_1 with the probability ε_1. The algorithm β_1 is said to (t_1, ε_1)-break the DLP. Let (g, $g^{\alpha} \bmod p$) be a random instance of the DLP. The objective of the algorithm β_1 is to derive α. In this proof, β_1 simulates challenger C in the game of Definition 1. In the meantime, $\mathcal{A}$ adaptively issues queries as those defined in the game of Definition 1.

– Signature-encryption queries: When $\mathcal{A}$ issues a Signature-encryption query on a message m, the algorithm β_1 first randomly chooses an integer $k \in Z_q^*$ and computes $C = (\beta^{h(y_b^*, ID_b)} h(ID_b) y_b^*)^k \bmod p$. Then, β_1 computes $r_1 = mh(C)^{-1} \bmod p$, $r_2 = h(m, h(g^k \bmod p), C) \bmod q$, and $s = k(1 + x_a r_2)^{-1} \bmod q$. Here, ($r_1$, r_2, s) is the authenticated ciphertext σ which is returned as the result of the Signature-encryption on the message m.
– Signature-verification queries: When $\mathcal{A}$ issues a Signature-verification query on an authenticated ciphertext σ = (r_1, r_2, s), β_1 first computes $K = g^s (\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{sr_2} \bmod p$ and $C' = K^{x_b^*} \bmod p$. Then, β_1 recovers $m = h(C') r_1 \bmod p$. If the recovered m is consistent with the redundancy check and the equality of $r_2 = h(m, h(K), C') \bmod q$ holds, β_1 outputs m; otherwise, the $\perp$ symbol is returned as a result.
– Signature-conversion queries: When $\mathcal{A}$ issues a Signature-conversion query on an authenticated ciphertext σ = (r_1, r_2, s), the algorithm β_1 first computes $K = g^s (\beta^{h(y_a, ID_a)} h(ID_a) y_a)^{sr_2} \bmod p$ and $C' = K^{x_b^*} \bmod p$. Then, β_1 recovers $m = h(C') r_1 \bmod p$. If the result (r_2, s, C') satisfies $r_2 = h(m, h(K), C') \bmod q$, outputs the result; Otherwise, the $\perp$ symbol is returned as result.
– Recipient-proof queries: When $\mathcal{A}$ issues a Recipient-proof query on an authenticated ciphertext σ = (r_1, r_2, s), β_1 first performs the same steps as those in Signature-conversion queries, and then chooses an integer e to compute $E = K^e \bmod p$, $W = E^{x_b^*} \bmod p$, and $W' = C'^{e} \bmod p$. If $W = W'$, β_1 outputs the symbol 1 as the result. Otherwise, the $\perp$ symbol is returned as result.

Challenge: The adversary $\mathcal{A}$ generates two messages, m_0 and m_1, of the same length. The challenger β_1 flips a coin $\lambda \leftarrow \{0, 1\}$ and computes an authenticated ciphertext σ^* = Signature-encryption (m_λ, x_a, y_b^*). The algorithm β_1 first randomly chooses an integer Z

Copyright ⓒ 2014 KSII

$\in Z_q^*$ and computes $C^* = (\beta^{h(y_b^*, ID_b)} h(ID_b) y_b^*)^Z \bmod p$. Then, β_1 computes $r_1^* = m_\lambda h(C)^{-1} \bmod p$, $r_2^* = h(m_\lambda, h(g^\alpha \bmod p), C) \bmod q$, and $s^* = Z(1 + x_a r_2)^{-1} \bmod q$. The authenticated ciphertext $\sigma^* = (r_1^*, r_2^*, s^*)$ is sent to $\mathcal{A}$ as the target challenge. If $Z = \alpha$, then σ^* is indeed a random Signature-encryption of m_λ. If Z is a random integer and does not equal to α, then r_1^* and s^* are random elements. Therefore, σ^* is independent of λ.

Phase 2: The adversary $\mathcal{A}$ issues new queries as those in Phase 1. It is not allowed to make a Signature-verification or Signature-conversion query for the target challenge σ^*.

Analysis: Consider the case when $Z = \alpha$, the distribution of the adversary $\mathcal{A}$'s view in the simulation is identical to that $\mathcal{A}$ is playing the game with C. Consequently, $\Pr_{\beta_1}[\text{Succ}] = \Pr_{\mathcal{A}}[\text{Succ}] - 1/2$, where $\Pr_{\mathcal{A}}[\text{Succ}]$ stands for the probability that $\mathcal{A}$ succeeds. When Z is uniformly distributed in Z_q^*, the adversary $\mathcal{A}$ has no information about the value of λ and hence the probability of $\lambda' = \lambda$ is at most 1/2. Therefore, we conclude that $\Pr_{\beta_1}[\text{Succ}] = \varepsilon_1 \geq \Pr_{\mathcal{A}}[\text{Succ}] - 1/2 = \varepsilon - 1/2$.

Q.E.D.

Theorem 8. *The proposed self-certified CAE scheme is* (t, ε)*-secure against existential forgery under chosen plaintext attacks if there exists no polynomial-time algorithm* β_2 *that can forge the Nyberg-Rueppel signature in time* t_2 *with the probability* ε_2.

Proof. Suppose that $\mathcal{F}$ is a (t, ε)-algorithm that breaks the self-certified CAE scheme under chosen message attacks in time t with the probability ε. In fact, the signature verification (/message recovery) part of the proposed scheme is based on and expanded from the Nyberg-Rueppel signature scheme. We will construct a (t_2, ε_2)-algorithm β_2 that forges the Nyberg-Rueppel signature in time t_2 with the probability ε_2 from the algorithm $\mathcal{F}$. The objective of the algorithm β_2 is to derive a valid Nyberg-Rueppel signature. In this proof, β_2 simulates $\mathcal{F}$'s challenger in the game of Definition 2 with the target signer U_a's public key y_a^* where $\beta^{h(y_a^*, ID_a)} h(ID_a) y_a^* = g^{x_a^*} \bmod p$. Then, $\mathcal{F}$ adaptively issues the same queries as those defined in the game of Definition 1.

Forgery: The algorithm $\mathcal{F}$ generates an authenticated ciphertext $\sigma^* = (r_1^*, r_2^*, s^*)$ for one target message m^* under the private key of the designated recipient. Note that σ^* is not obtained from a Signature-encryption query (m^*, x_a^*, y_b).

Analysis: $\mathcal{F}$ outputs an authenticated ciphertext σ^* which can be converted to the message m^* and its corresponding signature (r_2^*, s^*, C'^*) with a non-negligible probability. If (r_2^*, s^*, C'^*) satisfies the signature verification equation $r_2^* = h(m^*, h(K^*), C'^*) \bmod q$ with the probability ε, then (r_2^*, s^*, C'^*) can be regarded as a valid Nyberg-Rueppel signature of the message m^* with respect to the public key y_a^*. It can be seen that $\Pr_{\beta_2}[\text{Succ}]$ is hence at least $\Pr_{\mathcal{F}}[\text{Succ}]$. We conclude that $\Pr_{\beta_2}[\text{Succ}] = \varepsilon_2 \geq \varepsilon$.

Q.E.D.

4.3 AVISPA

We show through the simulation for the formal security verification using the widely accepted AVISPA (automated validation of internet security protocols and applications) tool [34] that our schemes are secure against malicious attacks. This is a push button tool for the automated validation of security protocols. There is a modular and expressive formal language called HLPSL (high level protocols specification language) which is used by AVISPA to specify the security protocol and their properties. It is a role-based language; therefore, we must first determine the sequence of actions of each kind of protocol participants in a module. This specification can later be instantiated by one or more agents playing the given role, and we further specify how the resulting participants interact with one another by combining multiple basic roles together into a composed role. HLPSL specification is translated into the Intermediate Format (IF), using hlpsl2if. IF specification is then processed by model-checkers

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfile03FD9u.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.06s
  visitedNodes: 8 nodes
  depth: 3 plies
```

Fig. 2. The result of the analysis using OFMC on our scheme

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfilealvjGr.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.01 seconds
  Computation: 0.00 seconds
```

Fig. 3. The result of the analysis using ATSE on our scheme

to analyze if the security goals are violated. There are four different verification back end tools use to analyze the IF specification namely, OFMC (On-the-Fly Model-Checker), CL-AtSe (Constraint Logic based Attack Searcher), SATMC (SAT-based Model Checker), TA4SP (Tree Automata based Protocol Analyser). We refer to the samples of [35] for a detailed description of HLPSL. **Fig. 2** and **3** are outputs of running OFMC and ATSE tools [36] on our proposed scheme I based on discrete logarithms.

4.4 Performance Evaluation

In this subsection, we compared the performance among previously proposed schemes [15, 17] and our DLP ones stated in Section 2. For assuring the validity of recipient's public key, the Wu-Hsu scheme [15] has to perform extra public key verification before any cryptographic mechanisms. On the contrary, since Lv *et al.*'s scheme [17] and our proposed ones adapt the properties of self-certified public key systems, the tasks of verifying the signature and authenticating the public key can be achieved simultaneously, which benefits

Copyright ⓒ 2014 KSII

the transmission and computation performance. Since the modular exponentiation computation is the most time-consuming operation, we ignore others such as the one-way hash, multiplication, inverse and addition operation. As the detailed comparisons shown in **Table 4**, it can be seen that the proposed schemes not only preserve the merit that the signature conversion process requires neither computation efforts nor communication overheads, but also outperform the Wu-Hsu scheme and Lv *et al.*'s scheme in terms of the computation efficiency for the entire protocol. Note that the Wu-Hsu scheme [15] has to perform extra public key verification and does not provide the property of recipient proof. Seeing that Lv *et al.*'s scheme will incur rather high computation complexity, we proposed six efficient variants based on the ElGamal signature scheme. Specifically, we optimize the signer's computational cost in variants II, IV and V, i.e., $3T_e$. Moreover, the last column in **Table 4** shows that the proposed schemes are more efficient than the others.

5. Conclusions

In some applications, the signature only needs to be verified by some specified recipients while keeping the message secret from the public. The authenticated encryption schemes can be used to achieve this purpose. In the normal procedure, only the recipient can recover the message and verify the signature. In case that the signer repudiates his signing, the recipient can release an ordinary signature for public verification. In this paper, we have proposed

Table 4. Comparisons among previously proposed schemes and our DLP ones.

Scheme	Signature generation	Message recovery and signature verification	Converted signature verification	Recipient proof	Total cost
WH [15]	$5T_e$	$6T_e$	$5T_e$	N.A.	$16T_e$
LWK[17]	$5T_e$	$6T_e$	$3T_e$	$7T_e$	$21T_e$
I	$3T_e$	$4T_e$	$3T_e$	$6T_e$	$16T_e$
II	$3T_e$	$3T_e$	$3T_e$	$6T_e$	$15T_e$
III	$3T_e$	$4T_e$	$3T_e$	$6T_e$	$16T_e$
IV	$3T_e$	$3T_e$	$3T_e$	$6T_e$	$15T_e$
V	$3T_e$	$3T_e$	$3T_e$	$6T_e$	$15T_e$
VI	$3T_e$	$4T_e$	$3T_e$	$6T_e$	$16T_e$

Remarks:
1. Let T_e be the time for performing a modular exponentiation computation.
2. WH and LWK separately represent the Wu-Hsu and Lv *et al.*'s schemes. I to VI denote the proposed schemes I to VI, respectively.
3. Suppose that verifying the public key certificate of the Wu-Hsu scheme is implemented with ElGamal signature verification [6], i.e., $T_m + 3T_e$.

efficient and computationally indistinguishable self-certified CAE schemes based on discrete logarithms along with their elliptic curve variants. Implemented with self-certified public key systems, our proposed schemes require no extra certificate since the tasks of verifying the signature and authenticating the public key can be simultaneously carried out in one step.

In case of a later dispute, the recipient has the ability to solely convert the signature into an ordinary one without any computation efforts or communication overheads. In addition, the recipient is capable of convincing anyone that he is the real recipient. The proposed schemes are shown to be efficient and secure against known existential active attacks. Furthermore, compared with existing CAE schemes [15, 17], our scheme greatly reduces the computational costs. They also satisfy the semantic security requirement. Moreover, the elliptic curve variants with shorter key length can help with faster execution and more bandwidth savings, so as to provide crucial benefits in the applications of limited computing power and insufficient storage space like mobile phones, etc.

Acknowledgement

This work was supported in part by the National Science Council of Republic of China under the contract number NSC 102-2221-E-019-041.

References

[1] W. Stallings, *Cryptography and network security: principles and practices*, 3rd. Ed., Prentice Hall, 2002.
[2] F. Hou, Z. Wang, Y. Tang and Z. Liu, "Protecting integrity and confidentiality for data communication," *Proceedings of Ninth International Symposium on Computers and Communications (ISCC)*, vol. 1, no. 28, pp. 357-362, 2004. Article (CrossRef Link)
[3] B. Meng, S. Wang and Q. Xiong, "A fair non-repudiation protocol," in *Proc. of the 7th International Conference on Computer Supported Cooperative Work in Design*, pp. 68-73, 2002. Article (CrossRef Link)
[4] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644-654, 1976. Article (CrossRef Link)
[5] A. Menezes, P. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc, 1997.
[6] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, no. 4, pp. 469-472, 1985. Article (CrossRef Link)
[7] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978. Article (CrossRef Link)
[8] ISO/IEC 9594-8, Information technology – open systems interconnection – the directory: public-key and attribute certificate frameworks, International Organization for Standardization, 2001.
[9] A. Shamir, "Identity-based cryptosystems and signature schemes," *Advances in Cryptology – CRYPTO'84*, Springer-Verlag, pp. 47-53, 1984. Article (CrossRef Link)
[10] M. Girault, "Self-certified public keys," *Advances in Cryptology – EUROCRYPT'91*, Springer-Verlag, pp. 491-497, 1991. Article (CrossRef Link)
[11] VISA and MasterCard Inc, Secure Electronic Transaction (SET) Specification, Version 1.0, 1997.
[12] P. Horster, M. Michel and H. Peterson, "Authenticated encryption schemes with low communication costs," *Electronics Letters*, vol. 30, no. 15, pp. 1212-1213, 1994. Article (CrossRef Link)
[13] E. J. Yoon and K. Y. Yoo, "Robust authenticated encryption scheme with message linkages," in *Proc. of Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES)*, pp. 281-288, 2005. Article (CrossRef Link)
[14] S. Araki, S. Uehara and K. Imamura, "The limited verifier signature and its application," *IEICE*

Copyright ⓒ 2014 KSII

Transactions on Fundamentals, vol. E82-A, no. 1, pp. 63-68, 1999. Article (CrossRef Link)
[15] T. S. Wu and C. L. Hsu, "Convertible authenticated encryption scheme," *The Journal of Systems and Software*, vol. 62, no. 3. pp. 205-209, 2002. Article (CrossRef Link)
[16] Y. H. Chen and J. K. Jan, "Enhancement of digital signature with message recovery using self-certified public keys and its variants," *ACM SIGOPS Operating Systems Review*, vol. 39, no. 3, pp. 90-96, 2005. Article (CrossRef Link)
[17] 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, pp. 1285-1297, 2005. Article (CrossRef Link)
[18] T. S. Wu and H. Y. Lin, "Efficient self-certified proxy CAE scheme and its variants," Journal of Systems and Software, vol. 82, no. 6, pp. 974-980, 2009. Article (CrossRef Link)
[19] Q. Xie, G. Wang, F. Xia, and D. Chen, "Self-certified proxy convertible authenticated encryption: formal definitions and a provably secure scheme," *Concurrency and Computation: Practice and Experience*, 2013. Article (CrossRef Link)
[20] C. L. Hsu and H. Y. Lin, "New identity-based key-insulated convertible multi-authenticated encryption scheme," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1724-1731, 2011. Article (CrossRef Link)
[21] J. L. Tsai, N. W. Lo, T. C. Wu, "Efficient convertible multi-authenticated encryption scheme for group communications," *Biometrics and Security Technologies (ISBAST)*, pp. 54-58, 2012. Article (CrossRef Link)
[22] ANSI X9.31, Digital signatures using reversible public key cryptography for the financial services industry (rDSA), 1998.
[23] ANSI X9.62, Public key cryptography for the financial service industry - the elliptic curve digital signature algorithm (ECDSA), Draft, 1998.
[24] ANSI X9.63, Public key cryptography for the financial services industry - key agreement and key transport using elliptic curve cryptography, 2001.
[25] IEEE P1363, Standard specifications for public key cryptography, *The Institute of Electrical and Electronics Engineers*, Inc., 2000.
[26] ISO/IEC 14888-3, Information technology – security techniques – digital signature with appendix – part 3: certificate-based mechanisms, International Organization for Standardization, 1998.
[27] ISO/IEC 15946-3, Information technology – security techniques – cryptographic techniques based on elliptic curves – part 3: key establishment, International Organization for Standardization, 2002.
[28] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987. Article (CrossRef Link)
[29] V. Miller, "Use of elliptic curves in cryptography," *Advances in Cryptology – CRYPTO'85*, Springer-Verlag, pp. 417-426, 1985. Article (CrossRef Link)
[30] I. Blake, G. Seroussi and N. Smart, "Elliptic curves in cryptography," *London Mathematical Society Lecture Note Series 265*, Cambridge University Press, 1999. Article (CrossRef Link)
[31] A. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993. Article (CrossRef Link)
[32] K. Nyberg and R. A. Rueppel, "A new signature scheme based on the DSA giving message recovery," in *Proc. of the 1st ACM Conference on Computer and Communication Security*, ACM Press, pp. 58-61, 1993. Article (CrossRef Link)
[33] K. Nyberg and R. A. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem," *Advances in Cryptology – EUROCRYPT'94*, Springer-Verlag, pp. 182-193, 1994. Article (CrossRef Link)
[34] AVISPA. Automated validation of internet security protocols and applications. http://www.avispa-project.org/. Accessed on January 2014.
[35] A. K. Das and B. Bruhadeshwar, "An improved and effective secure password-based authentication and key agreement scheme using smart cards for the telecare medicine

information system," *Journal of Medical Systems*, vol. 37, no. 5, pp. 1 - 17, 2013. Article (CrossRef Link)
[36] AVISPA. AVISPA web tool. http://www.avispa-project.org/web-interface/expert.php/. Accessed on January 2014.

Tzong-Sun Wu received his B.S. degree in Electrical Engineering from National Taiwan University in 1990 and his Ph.D. in Information Management from National Taiwan University of Science and Technology in 1998. From August 1998 to July 2001, he has been an Assistant Professor in Department of Information Management, Huafan University. From August 2001 to January 2007, he has been an Associate Professor in Department of Informatics, Fo Guang University. He is currently with Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan. His research interests include information security, watermarking, digital right management, and e-commerce.

Yih-Sen Chen received his B.A. degree in Information Management from Aletheia University, Taiwan in 1997 and his M.S. degree in Informatics from Fo Guang University of Taiwan in 2006. Now he is a doctoral candidate in the Department of Computer Science and Engineering, National Taiwan Ocean University. His main research interests include cryptography, information theory, security management, and network security.

Han-Yu Lin received his Ph.D. degree in computer science and engineering from the National Chiao Tung University, Taiwan in 2010. He served as a part-time Assistant Professor in both the Department of Information Management, Chang Gung University, Taiwan and the Department of Information Management, Kainan University, Taiwan from 2011. He was an engineer in CyberTrust Technology Institute, Institute for Information Industry, Taiwan from January 2012 to July 2012. Since August 2012, he has been an Assistant Professor in the Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan. His research interests include Cryptology, Network Security, Digital Forensics, RFID Privacy and Application, Cloud Computing Security and E-commerce Security.

Appendix

```
role alice (    A,B : agent,
                Kab : symmetric_key,
                H   : hash_func,
        SND, RCV : channel(dy) )
played_by A
def=
                G          : text,
                M          : message,
                Ya, Yb     : public_key,
                MUL        : hash_func
const alice_bob_na, bob_alice_nb : protocol_id
init State := 0
```

Copyright ⓒ 2014 KSII

```
    local   State     : nat,
            R1,R2     : text,
            K,C       : text,
            S         : text,
exp(MUL(exp(H(Ya.Yb),H(Yb.B)).H(B).Yb),K')
            /\ R1' := MUL(M.H(C'))
            /\ R2' := H(M.H(exp(G, K').C'))
            /\ S'  := MUL(K'.inv(Ya).R2')
            /\ SND({R1'.R2'.S'}_Kab)
            /\ witness(A,B,bob_alice_nb,R2')
end role
```

```
role bob (   A,B : agent,
             Kab : symmetric_key,
             H   : hash_func,
      SND, RCV : channel(dy))
played_by B
def=
    local    State   : nat,
         R1,R2, S    : text,
             K,C     : text,
               G     : text,
               M     : message,
          Ya, Yb     : public_key,
         MUL         : hash_func
init State := 1
transition
1. State    = 1 /\ RCV({R1'.R2'.S'}_Kab)   =|>
    State'  := 3 /\ K' :=
MUL(exp(G,S').exp(MUL(exp(H(Ya.Yb),H(Ya.A)).H
(A).Ya),MUL(S'.R2')))
            /\ C'   := exp(K',inv(Yb))
            /\ M'   := MUL(H(C').R1')
            /\ R2'  :=H(M'.H(K').C')
            /\ witness(B, A, alice_bob_na, R2')
            /\ request(B, A, bob_alice_nb, R2')
end role
```

```
transition
1. State    = 0 /\ RCV(start) =|>
    State' := 2 /\ K'   := new()
                /\ C'   :=
role session(A,   B : agent,
             Kab   : symmetric_key,
             Hash  : hash_func)
def=
    local SA , RA,
          SB , RB   : channel (dy)
composition
       alice (A, B, Kab, Hash, SA, RA)
    /\ bob   (A, B, Kab, Hash, SB, RB)
end role
```

```
role environment()
def=
    const a, b        : agent,
      kab, kai, kib   : symmetric_key,
      h               : hash_func,
      bob_alice_nb, k : protocol_id
intruder_knowledge = {a, b, h, kai, kib}

composition
    session(a, b, kab, h)
  /\ session(a, i, kai, h)
  /\ session(i, b, kib, h)
end role

goal
secrecy_of k
authentication_on bob_alice_nb
end goal

environment()
```