Practical Convertible Authenticated Encryption Schemes Using Self-certified Public Keys

Jiqiang Lv [a] , Xinmei Wang [a] and Kwangjo Kim [b]

[a] *National Key Lab of ISN,*
Xidian University,
Xi'an City, Shaanxi Province, 710071 CHINA
lvjiqiang@hotmail.com,xmwang@xidian.edu.cn

[b] *International Research center for Information Security,*
Information and Communications University,
58-4 Hwaam-dong Yusong-ku, Taejon, 305-732 KOREA
kkj@icu.ac.kr

Abstract

A convertible authenticated encryption scheme allows a designated receiver to recover and verify a message simultaneously, during which the recipient can prove the dishonesty of the sender to any third party if the sender repudiates her signature later. In this paper, after showing some weaknesses in Wu *et al.*'s [21] and Huang *et al*'s [10] convertible authenticated encryption schemes, we propose a practical convertible authenticated encryption scheme using self-certified public keys and then extend it to one with message linkages when the signing message is large. Each scheme could provide semantic security of the message, the signer's public key can be simultaneously authenticated in checking a signature' validity and only under the cooperation of the recipient could a verifier know to whom a specific signature is sent. Finally, we give a variant that could make a verifier know to whom a signature is sent while verifying its validity.

Key words: Public key cryptology; Authenticated encryption scheme; Self-certified public key; Message linkages
PACS:

1 Introduction

`Convertible Authenticated Encryption` A digital signature provides the functions of integration, authentication and nonrepudiation for a signing message. However, in some situations, a signature only needs to be verified by some specified recipients while keeping the message secret from the public. By modifying Nyberg *et al.*'s message recovery signature[14], Horster *et al.* [8] firstly proposed an authenticated encryption scheme with the above property. Since then, some similar schemes have been proposed [23,24,12,9,15,20].

However, since no one except the specified recipient can be convinced of the signer's signature in an authenticated encryption scheme, so if the signer repudiates her signature later, it cannot make the recipient prove the dishonesty of the signer to any verifier without releasing his secret. To overcome this weakness, Araki *et al.* [2] proposed a convertible limited verifier scheme to enable the recipient to convert the signature to an ordinary one so that any verifier can verify its validity. But it needs the cooperation of the signer when the recipient converts the signature, which is obviously a weakness under the situation that the signer is unwilling to cooperate.

Recently, Wu *et al.* [21] proposed a convertible authenticated encryption scheme. During which, the recipient can easily produce the ordinary signature without the cooperation of the signer, and if the signer wants to repudiate her signature, he can reveal the converted signature and then any verifier can prove the dishonesty of the signer. Unfortunately, Huang *et al.* [10] showed that Wu *et al.*'s scheme does not consider that once an intruder knows the message then he can also easily convert a signature into an ordinary one and claim that the signature is sent to him. Finally, they proposed a new convertible authenticated encryption scheme to solve this problem.

On the other hand, if the signing message is large, the message must be divided into a sequence of small message blocks and each message block can be encrypted and signed as a signature block individually. But this approach has a weakness that an intruder can reorder or partially delete blocks so that the recipient cannot realize this. To overcome this weakness and reduce communication costs, some schemes with message linkages have been proposed[11,13,18,20].

`Self-Certified Public Keys` Since the signer's public key must be used to verify a digital signature in a public key cryptosystem, it is necessary to check the public key's correctness before proceeding to the signature verification. Girault [7] firstly introduced the notion of self-certified public keys, during which each user's public keys is derived from the signature of the user's identity with his secret key that is chosen by the user himself but created by the system authority. The public key of each user need not be companied with a separate certificate to be authenticated by verifiers. The authentication of the public key can implicitly be accomplished with the signature verification.

By using self-certified public keys, the system authority need not maintain the public keys and the certificate directory, thus can reduce the amount of storage and computation cost.

`Our Contribution` In this paper, we firstly show that either Wu *et al.*'s or Huang *et al.*'s scheme cannot provide semantic security for the message, that is, any adversary can determine whether his guessed message is the actual message signed by the original signer after he gets a valid signature. Furthermore, Huang *et al.*'s scheme has another weakness: once an adversary gets a valid signature on a specific message, then he can recover another message if he gets its corresponding signature.

Following, we propose a convertible authenticated encryption scheme using self-certified public keys, and then extend it to one with message linkages when the signing message is large. Each scheme provides semantic security of the message, i.e., after getting a valid signature, any adversary cannot determine whether his guessed message is the actual message; If the signer repudiates her signature later, then without the cooperation of the signer, the recipient can prove the dishonesty of the signer to any verifier by revealing the message and its converted signature; If the recipient does not reveal the converted signature, any verifier cannot check the message's validity even though he gets its corresponding signature; A verifier can not know to whom a signature is sent while verifying its validity. Only under the cooperation of the recipient could a verifier determine whether a signature is sent to the recipient.

We also give a variant during which a verifier could know to whom a signature is sent while verifying its validity.

`Organization of the Paper` The rest of the paper is organized as follows. In the next section, we briefly show some weaknesses in Wu *et al.*'s and Huang *et al*'s convertible authenticated encryption schemes, respectively. In Section 3, we present a convertible authenticated encryption scheme using self-certified public keys, then extend it to a scheme with message linkages, and finally give a variant. In Section 4, we make a simple security analysis and computational complexity of the proposed schemes. A conclusion is made in Section. 5.

2 Weaknesses in Wu *et al.*'s and Huang *et al*'s Convertible Authenticated Encryption Schemes

Let's firstly list some notation and parameters that will be used in this section only: Let p, q be two public large primes with $q|p-1$, g be a public generator of order q in Z_p and $H(\cdot)$ be a public one-way hash function. (x_a, y_a) is the signer *Alice*'s secret and public keys, where $y_a = g^{x_a} \bmod q$. (x_b, y_b) is the recipient *Bob*'s secret and public keys, where $y_b = g^{x_b} \bmod q$.

2.1 Wu et al.*'s Scheme [21]*

To produce the signature for M, the signer *Alice* first chooses an integer k from Z_q^*, and computes $r_1 = M \cdot (H(y_b^k \bmod p)^{-1}) \bmod p, r_2 = H(M, H(g^k \bmod p)^{-1}) \bmod q, s = k - r_2 \cdot x_a \bmod q$, Finally, she sends the triple (r_1, r_2, s) to the recipient *Bob.*

Bob first recovers the message as $M = H((g^s \cdot y_a^{r_2})^{x_b} \bmod p) \cdot r_1 \bmod p$, and checks if $r_2 = H(M, H(g^s \cdot y_a^{r_2} \bmod p)) \bmod q$. If it holds, then the signature is valid.

Later on, if the signer *Alice* repudiates the signature, *Bob* can prove the dishonesty of *Alice* by revealing the converted signature (r_2, s) for message M. With this converted signature, anyone can verify its validity with the equation $r_2 = H(M, H(g^s \cdot y_a^{r_2} \bmod p)^{-1}) \bmod q$.

Weakness Suppose an adversary gets a valid (r_1, r_2, s), he can check whether his guessed message M^* satisfies $r_2 = H(M^*, H(g^s \cdot y_a^{r_2} \bmod p)) \bmod q$. If it holds, then he gets the actual message. So Wu *et al.*'s scheme cannot provide the semantic security of the message.

2.2 Huang et al.*'s Scheme [10]*

To produce the signature for M, the signer *Alice* randomly chooses an integers k from Z_q^*, and computes $c = M \cdot y_b^{q-k} \bmod p$, $r = H(M, y_b, g^k) \bmod q$, and $s = k - r \cdot x_a \bmod q$. Finally, she sends the triple (c, r, s) to the recipient *Bob.*

Bob first recovers the message as $M = c \cdot (y_a^r \cdot g^s)^{x_b} \bmod p$ and checks $r \overset{?}{=} H(M, y_b, y_a^r \cdot g^s) \bmod q$. If it holds, then the signature is valid.

Later on,if the signer *Alice* repudiates the signature, *Bob* can prove the dishonesty of *Alice* by revealing the converted signature (r, s) for message M. With this converted signature, anyone can verify its validity with equation $r = H(M, y_b, y_a^r \cdot g^s) \bmod q$. Note that there is *Bob*'s public key y_b in the verification equation, so any verifier can be convinced that the signature is sent to *Bob.*

Weakness 1 Suppose an adversary gets a valid (c, r, s), he can determine whether his guessed message M^* is the actual message by checking if M^* satisfies $r = H(M^*, y_b, y_a^r \cdot g^s) \bmod q$. So Huang *et al.*'s scheme cannot provide the semantic security of the message, too.

Weakness 2 Suppose that the adversary has gotten a valid signature (c_1, r_1, s_1)

on message M_1, then he can compute $y_a^{x_b} = (M_1 \cdot c_1^{-1} \cdot y_b^{-s_1})^{r_1^{-1}} \bmod p$ from $M_1 = c_1 \cdot (y_a^{r_1} \cdot g^{s_1})^{x_b} \bmod p$. Now if he gets another valid signature (c_2, r_2, s_2) on message M_2, he can recover the message M_2 as $M_2 = c_2 \cdot (y_a^{x_b})^{r_2} \cdot y_b^{s_2} \bmod p$. So Huang *et al.*'s scheme is insecure.

3 Proposed Practical Convertible Authenticated Encryption Schemes Using Self-certified Public Keys

Semantic security is of very importance to an authenticated encryption scheme for practical communications. Otherwise, if the possible messages are limited, then an adversary can eventually determine which message the signer signs by checking which satisfies the verification equalities.

Under some real situations, the recipient may hope that a verifier does not know a signature is sent to him while checking its validity, but he may hope that he could prove this if he wants. Therefore, after he is convinced that exposing that he is the real recipient will benefit himself, he will prove that a signature is really sent to him, otherwise, he will not and just keep silent. While under other situations, the recipient may hope a verifier explicitly knows a signature is sent to him.

In this section, we propose a basic convertible authenticated encryption scheme using self-certified public keys, and then extend it to a scheme with message linkages when the signing message is large. During these two schemes, a verifier cannot know to whom a signature is sent while checking its validity. At the end of this section, we present a variant, during which a verifier could know to whom a signature is sent while checking its validity.

3.1 A Basic Convertible Authenticated Encryption Scheme

The basic scheme consists of the following five phases: system initialization, signature generation, signature recovery and verification, conversion and recipient proof.

System Initialization

The trusted authority, *TA*, chooses two large and distinct primes p^*, q^*, forms the other two large primes $p = 2p^* + 1$ and $q = 2q^* + 1$, and computes $n = p \cdot q$. Then, *TA* selects a generator g in Z_n, where g has an order of p^*q^*, and a public one-way hash function $H(\cdot)$. *TA* publishes n, g and $H(\cdot)$ to all users and keeps (p^*, q^*, p, q) secret.

When a user, *Alice* say, intends to join the system, she first chooses a secret key x_a and computes $p_a = g^{x_a} \bmod n$. Then she sends p_a and her identity ID_a to *TA*. After receiving them, TA computes $y_a = (p_a - ID_a)^{H(ID_a)^{-1}} \bmod n$ as *Alice*'s public key. Alice can check the validity of y_a by verifying the equation $y_a^{H(ID_a)} + ID_a = g^{x_a} \bmod n$. Every participant in this cryptosystem must register in the same way.

Signature Generation

To sign a message $M \in Z_n$ to a recipient *Bob*, *Alice* does the following[1],

Step 1: *Alice*, who knows the identity ID_b and the public key y_b corresponding to the secret key x_b of a recipient, *Bob*, randomly selects an integer x, and computes

$$\begin{aligned} r &= M \cdot (y_b^{H(ID_b)} + ID_b)^{-x} \bmod n, \\ v &= g^{x \cdot (y_b^{H(ID_b)} + ID_b)^{x_a}} \bmod n, \\ c &= H(M, v, g^x), \\ s &= x - c \cdot x_a. \end{aligned} \tag{1}$$

Step 2: *Alice* sends the tuple (c, r, s) to the recipient *Bob*.

Message Recovery and Verification

After receiving the tuple (c, r, s), the recipient *Bob*, computes

$$\begin{aligned} Y_a &= y_a^{H(ID_a)} + ID_a \bmod n, \\ M &= r \cdot (g^s \cdot {Y_a}^c)^{x_b} \bmod n, \\ v &= (g^s \cdot {Y_a}^c)^{{Y_a}^{x_b}} \bmod n. \end{aligned}$$

Then, *Bob* checks if the following equation holds:

$$c = H(M, v, g^s \cdot {Y_a}^c). \tag{2}$$

If it holds, then he is convinced that the signature is a valid signature from *Alice*. Rejects, otherwise.

Conversion

[1] During the scheme as well as the following schemes, we assume that *Alice* and *Bob* will keep $g^{x_a \cdot x_b}$ secret, which can be regarded as a long term session key between them.

If the signer *Alice* wants to repudiate her signature later, the recipient *Bob* can prove *Alice*'s dishonesty to any verifier by revealing the message M and the parameter v for a given (c, s). Any verifier can check *Alice*'s dishonesty by Eqn. (2). Only if it holds does the verifier accept the signature is generated by *Alice*. If *Bob* does not reveal v, any verifier cannot check the validity of the message even though he gets the message M and the corresponding signature (c, r, s).

Recipient Proof

If *Bob* wants to prove to any verifier *Tom* that he is the real recipient, they can do as follows:

Step 1: *Bob* first sends the message M, the parameter v and the signature (c, s) to *Tom*.

Step 2: After determining *Bob*'s identity, *Tom* computes

$$Y_a = y_a^{H(ID_a)} + ID_a \bmod n.$$

and then checks if Eqn. (2) holds. If it holds, then he continues the following steps. Otherwise, terminates the protocol.

Step 3: *Tom* selects a random integer k, computes

$$K = (g^s \cdot Y_a^c)^k \bmod n$$

and then sends K to *Bob*;

Step 4: After receiving K, *Bob* computes $Z = K^{Y_a^{x_b}} \bmod n$, and returns it to *Tom*.

Step 5: *Tom* computes $Z^* = v^k \bmod n$, and checks if $Z = Z^*$ holds. If it holds, then he is convinced that the signature is sent to *Bob*.

Theorem 1 *Given a valid signature (c, r, s), following the steps in the basic convertible authenticated encryption scheme, the recipient will surely recover and verify the message M from the signature.*
Proof: Since $Y_a = y_a^{H(ID_a)} + ID_a \bmod n$, therefore,

$$\begin{aligned} & r \cdot (g^s \cdot {Y_a}^c)^{x_b} \bmod n = r \cdot (g^s \cdot g^{c \cdot x_a})^{x_b} \bmod n \\ = & r \cdot (y_b^{H(ID_b)} + ID_b)^{s + c \cdot x_a} \bmod n = r \cdot (y_b^{H(ID_b)} + ID_b)^x \bmod n \\ = & M. \end{aligned}$$

Bob could also recover the parameter v, since $v = (g^s \cdot {Y_a}^c)^{{Y_a}^{x_b}} \bmod n = g^{x \cdot (y_b^{H(ID_b)} + ID_b)^{xa}} \bmod n$. Finally, he could verify the message by Eqn. (2).

Note that only *Alice* could generate such a signature that satisfies the above equation, so *Bob* can determine whether a signature is valid or not.

3.2 A Convertible Authenticated Encryption Scheme with Message Linkages

For data communications, when the signing message M is large, it must be divided into a sequence of small message blocks $\{M_1, M_2, \cdots, M_l\}$, $M_i \in Z_n$, $i = 1, 2, \cdots, l$. If each message block is encrypted and signed individually, it will require more computation and communication costs. To achieve computation and communication efficiency, we extend the basic scheme to a scheme with message linkages in this section.

The scheme also consists of five phases: system initialization, signature generation, signature recovery and verification, conversion and recipient proof. The system initialization phase is the same as that in the basic scheme, so we will just describe the left four phases in the following.

Signature Generation

Alice carries out the following steps to generate the signature blocks for the large message M.

Step 1: *Alice* lets $r_0 = 0$ and chooses a random integer t, then computes $r_i = M_i \times f(r_{i-1} \oplus t) \bmod n$ for $i = 1, 2, \cdots, l$, where $f(\cdot)$ is another public one-way hash function, and $\oplus$ denotes the exclusive OR operator.

Step 2: *Alice* selects a random integer x, and computes

$$\begin{aligned}
r &= t \cdot (y_b^{H(ID_b)} + ID_b)^{-x} \bmod n, \\
v &= g^{x \cdot (y_b^{H(ID_b)} + ID_b)^{xa}} \bmod n, \\
L &= H(M_1 \| M_2 \| \cdots \| M_l), \\
c &= H(L, v, g^x), \\
s &= x - c \cdot x_a,
\end{aligned} \tag{3}$$

where $\|$ denotes string concatenation.

Step 3: Finally, *Alice* sends $(c, r, s, r_1, r_2, \cdots, r_l)$ to *Bob*.

Message Recovery and Verification

After receiving the signature $(c, r, s, r_1, r_2, \cdots, r_l)$, *Bob* carries out the following steps to recover the message and verifies the signature by using his secret key x_b, *Alice*'s public key y_a and her ID_a:

Step 1: *Bob* computes

$$\begin{aligned} Y_a &= y_a^{H(ID_a)} + ID_a \bmod n, \\ v &= (g^s \cdot {Y_a}^c)^{{Y_a}^{x_b}} \bmod n, \\ t &= r \cdot (g^s \cdot {Y_a}^c)^{x_b} \bmod n. \end{aligned}$$

Step 2: *Bob* recovers the message block $\{M_1, M_2, \cdots, M_l\}$ as follows:

$$M_i = r_i \cdot f(r_{i-1} \oplus t)^{-1} \bmod n, (i = 1, 2, \cdots, l, r_0 = 0).$$

Step 3: *Bob* computes

$$L = H(M_1 \| M_2 \| \cdots \| M_l).$$

Then, *Bob* checks if the following equation holds:

$$c = H(L, v, g^s \cdot {Y_a}^c). \tag{4}$$

If Eqn.(4) holds, then he is convinced that the signature is a valid signature from *Alice*. Rejects, otherwise.

Conversion

If the signer *Alice* wants to repudiate her signature later, the recipient *Bob* can prove the dishonesty of *Alice* by revealing the message block $\{M_1, M_2, \cdots, M_l\}$ and the parameter v for a given (c, s). Any verifier can check if Eqn. (4) holds after computing $L = H(M_1 \| M_2 \| \cdots \| M_l)$. Only if it holds does the verifier accept the signature is generated by *Alice*.

Recipient Proof

If *Bob* wants to prove to any verifier *Tom* that he is the real recipient, they can do as follows:

Step 1: *Bob* first sends the message block $\{M_1, M_2, \cdots, M_l\}$, the parameter v and the signature (c, s) to *Tom*.

Step 2: After determining *Bob*'s identity, *Tom* computes

$$\begin{aligned} Y_a &= y_a^{H(ID_a)} + ID_a \bmod n, \\ L &= H(M_1 \| M_2 \| \cdots \| M_l), \end{aligned}$$

and then checks if Eqn. (4) holds. If it holds, then he continues the following steps. Otherwise, terminates the protocol.

The left steps are the same as the recipient proof phase in the basic scheme.

Theorem 2 *If a valid signature $(c, r, s, r_1, r_2, \cdots, r_l)$ is produced by the convertible authenticated encryption scheme with message linkages, the recipient will surely recover and verify the correct message M from the signature.*
Proof: Since $Y_a = y_a^{H(ID_a)} + ID_a \bmod n$ and $x = s + c \cdot x_a$, so we have $v = g^{x \cdot (y_b^{H(ID_b)} + ID_b)^{x_a}} \bmod n = (g^s \cdot {Y_a}^c)^{{Y_a}^{x_b}} \bmod n = v$.

Since $r = t \cdot (y_b^{H(ID_b)} + ID_b)^{-x} \bmod n$, therefore the recipient *Bob* can recover $t = r \cdot (g^s \cdot {Y_a}^c)^{x_b} \bmod n$ by using his secret key x_b. Next, he could recover the message M_i by computing $M_i = r_i \cdot f(r_{i-1} \oplus t)^{-1} \bmod n$, for $i = 1, 2, \cdots, l$, $(r_0 = 0)$. Consequently, he can compute $L = H(M_1 \| M_2 \| \cdots \| M_l)$, and check the validity of the signature by Eqn. (4).

3.3 Variant

During the above two schemes, if we replace the two equalities Eqns.(1) and (3) with the following two equalities Eqns.(5) and (6), respectively,

$$c = H(M, v, y_b, g^x), \tag{5}$$
$$c = H(L, v, y_b, g^x), \tag{6}$$

and correspondingly, Eqns.(2) and (4) will be the following Eqns.(7) and (8), respectively,

$$c = H(M, v, y_b, g^s \cdot Y_a^c), \tag{7}$$
$$c = H(L, v, y_b, g^s \cdot Y_a^c). \tag{8}$$

Obviously, any verifier could know to whom a signature is sent while verifying its validity, since y_b is used in the verification equalities.

4 Analysis of the Proposed Schemes

4.1 Security

The security of our schemes is based on the following three difficult problems:
Discrete logarithm modulo a composite(DLMC)[1]: Given a large composite n of two primes, p and q, a generator g over Z_n, and $y = g^x \bmod n$, it is computationally infeasible to derive x.
Factorization (FAC)[1]: Given a large composite n of two primes, p and q, it is computationally infeasible to find p and q.
Intractability of reversing a one-way hash function(OWHF)[4]: It is computationally infeasible to derive x from a given hashed value $H(x)$, or to find two different values x, x^* such that $H(x) = H(x^*)$.

Correctness: From Theorem (1) and (2), we can see the correctness of our schemes is sound.

Now, let's first consider the security in the basic scheme.

Unforgeability: The self-certified public keys' security is the same as that in [7]. Under the intractability of FAC, anyone except TA cannot get p^*, q^*, p and q from n. During the signature generation phase, anyone except the signer cannot generate a valid signature, since it needs the secret key x_a to complete the signature. Assume an intruder intends to reveal the secret key x_a from the equation $s = x - c \cdot x_a$. For a given signature (c, r, s), there is one more unknown parameter x in each equation $s = x - c \cdot x_a$. Since the intruder cannot compute x from $g^x = g^s \cdot Y_a^c \bmod n$ under the intractability of *DLMC*, so he cannot get the secret key x_a of the signer from the single equation. And every time when signing a signature, the parameter x will be different, so the number of secret parameters is always greater than the number of available equations. Therefore, the intruder cannot work successfully. If an adversary wants to directly forge a signature on some message that satisfies $c = H(M, v, g^s \cdot Y_a^c) \bmod n$, he must face the *DLMC* or *OWHF* problem. Any modification to the triple (c, r, s) will cause the inequality $c \neq H(M, v, g^s \cdot Y_a^c) \bmod n$ hold.

Confidentiality: Only by using the secret key x_b of the recipient could the message M be correctly recovered during the message recovery and verification phase. After an adversary gets the signature (c, r, s), he cannot guess the corresponding message M, since he can neither correctly compute the parameter v from the signature, nor could he express the parameter v with his guessed message M^*, the public parameters y_a, y_b or the signature (c, r, s). So our basic convertible authenticated encryption scheme provides semantic security of the message M. The reason for semantic security is that an adversary cannot get

$Y_b^{x_a}$ (or $Y_a^{x_b}$) even if he once gets a message M and its corresponding converted signature (c, s, v).

Undeniability: Under the intractability of *DMLC*, *FAC* and *OWHF*, anyone except the signer *Alice* cannot get a group of c, v, s and M that satisfy Eqn.(2). During the conversion phase, if *Bob* do not reveal the parameter v, any verifier cannot verify the validity of the signature, even he knows the message M and the signature (c, r, s). After *Bob* reveals M, v and (c, s), any third party can check its validity by checking Eqn.(2). By using Y_a, the verifier can determine whether a signature is created by *Alice*. Since only the signer *Alice* could create such a signature that satisfies Eqn.(2), so once she creates a valid signature, she cannot repudiate her signature creation against anyone.

The same security is with our scheme with message linkages. Note that it needs the secret key x_b to recover the parameter t, so anyone else cannot compute it, and cannot process further. If $(r_1, r_2, \cdots, r_l)$ is modified, deleted or replicated, then the recovered message block will be different, which will cause the recovered L is not equal to the original L. The signature will not pass the verification equations. Since t is protected in the one-way hash function $f(\cdot)$, an adversary cannot derive t from $f(r_{i-1} \oplus t) = r_i \cdot M_i^{-1} \bmod n$ after he gets one block M_i.

4.2 Computational Complexity

Let T_i denote the time for one inverse computation with modulo n, T_e denote the time for one exponentiation computation with modulo n, T_{nm} denote the time for one multiplication computation without modulo n, T_m denote the time for one multiplication computation with modulo n, T_h denote the time for executing the adopted one-way hash function in each scheme, and $|x|$ mean the bit length of an integer x.

Then let's show the computational complexity in each scheme[2]. The computational complexity in each phase of the basic scheme is as follows, signature generation phase is $3T_e + 2T_{nm} + T_h + T_m + T_i$, message recovery phase is $2T_e + 2T_m$, message verification phase is $T_e + T_h$, signature conversion phase is 0, conversion verification phase is $3T_e + 2T_h + T_m$ and recipient proof phase is $6T_e + 2T_h + T_m$.

The computational complexity in each phase of the scheme with message linkages is as follows, signature generation phase is $(l + 2)T_h + (l + 1)T_m + 3T_e +$

[2] We assume *Alice* precomputes $Y_b = y_b^{ID_b} + ID_b \bmod n$ and $Y_b^{x_a} \bmod n$ and *Bob* precomputes $Y_a = y_a^{ID_a} + ID_a \bmod n$ and $Y_a^{x_b} \bmod n$.

$2T_{nm} + T_i$, message recovery phase is $(l + 2)T_m + lT_h + lT_i + 2T_e$, message verification phase is $4T_e + 2T_h + T_m$, signature conversion phase is 0, conversion verification phase is $3T_e+3T_h+T_m$ and recipient proof phase is $6T_e+3T_h+T_m$.

Each variant has the same communication costs and computational complexity as its corresponding scheme except that it does not need a recipient proof phase.

5 Conclusion

After showing some weaknesses in Wu *et al.*'s [21] and Huang *et al*'s [10] convertible authenticated encryption schemes, we propose a convertible authenticated encryption scheme using self-certified public keys, so that the signer's public key can be simultaneously authenticated in checking a signature' validity. Then, we extend it to one with message linkages when the signing message is large. Each proposed scheme provides semantic security of the message, that is, after getting a valid signature, any adversary cannot determine whether his guessed message is the actual message by checking if it satisfies the verification equalities; Only under the cooperation of the recipient could a verifier know to whom a specific signature is sent. We also give a variant, during which a verifier could know to whom a signature is sent while verifying its validity.

References

[1] L. Adleman and K. McCurley, Open Problems in Number Theoretic Complexity, Proc. of the 1994 Algorithmic Number Theory Symposium, Springer-Verlag, LNCS 877:291-322(1994).

[2] S. Araki, S. Uehara and K. Imamura, The Limited Verifier Signature and Its Application, IEICE Transactions on Fundamentals, Vol. E82-A(1):63-68(1999).

[3] F. Bao and R.H. Deng, A Signcryption Scheme with Signature Directly Verifiable by Public Key, Proc. of PKC'98-Public Key Cryptography, Springer-Verlag, LNCS 1431:55-59(1998).

[4] W. Diffle and M. Hellman, New Directions in Cryptology, IEEE Transactions on Information Theory, IT-22(6):644-654(1996).

[5] Y. Dodis and J.H. An, Concealment amd Its Applications to Authenticated Encryption, Advance in Cryptology-EUROCRYPT'03, Springer-Verlag,LNCS 2656:312-329(2003).

[6] T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transactions on Information Theory, IT-30(4):469-472(1985).

[7] M. Girault, Self-certified public keys, Advance in Cryptology-EUROCRYPT'91, Springer-Verlag, LNCS 547:491-497(1991).

[8] P. Horster, M. Michels and H. Petersen, Authenticated Encryption Schemes with Low Communication Costs, IEE Electronics Letters, Vol. 30(15):1212-1213(1994).

[9] C. Hsu and T. Wu, Authenticated encryption schemes with (t, n) shared verification, IEE Proc.-Computer Digital Techology, Vol.145(2):117-120(1998).

[10] H. Huang and C. Chang, An Efficient Convertible Authenticated Encryption Scheme and its Variant, Proc. of ICICS2003-Fifth International Conference on Information and Communications Security, Springer-Verlag, LNCS 2836:382-392(2003).

[11] S. Hwang, C. Chang and W. Yang.: Authenticated encryption schemes with message linkages, Information Processing Letters, Vol.58(4):189-194(1996).

[12] W. Lee and C. Chang, Authenticated encryption schemes without using a one way function, IEE Electronics Letters, Vol.31(19):1656-1657(1995).

[13] W. Lee and C. Chang, Authenticated encryption schemes with linkage between message blocks, Information Processing Letters, Vol.63(5):247-250(1997).

[14] K. Nyberg and R.A. Rueppel, Message Recover for Signature Schemes Based on the Discrete Logarithm Problem, Advance in Cryptology-EUROCRYPT'94, Springer-Verlag,LNCS 950:182-193(1995).

[15] H. Petersen and M. Michels, Cryptanalysis and Improvement of Signcryption Schemes, IEE Proc.-Computers and Digital Techniques, Vol.145(2):149-151(1998).

[16] B. Schneier, Applied Cryptology, second edition, Wiley, New York, 1996.

[17] C.P. Schnorr, Efficient Identification and Signatures for Smart Cards, Advance in Cryptology-CRYPTO'89, Springer-Verlag, LNCS 435:339-351(1990).

[18] Y. Tseng and J. Jan, An efficient authenticated encryption scheme with message linkages and low communication costs, Journal of Information Science and Engineering, Vol.18(1):41-46(2002).

[19] Y. Tseng, J. Jan and H. Chien, Authenticated encryption schemes with messages for message flows, International Journal of Computers and Electrical Engineering, Vol.29(1):101-109(2003).

[20] Y. Tseng, J. Jan and H. Chien, Digiatal Signature with Message Recovery using Self-certified Public Keys and its Variant, Journal of Applied Mathematics and Computation. Vol.136:203-214(2003).

[21] T. Wu and C. Hsu, Convertible Authenticated Encryption Scheme. The Journal of Systems and Software, Vol.62:205-209(2002).

[22] F. Zhang and K. Kim, A Universal Forgery of Araki *et al.*'s Convertible Limited Verifier Signature Scheme, IEICE Trans. Fundamentals, Vol.E86-A(2):515-516(2003).

[23] Y. Zheng, Digital Signcryption or How to Achieve cost($signture + encryption$) $\ll$ cost($signature$) + $cost$($encryption$), Advance in Cryptology-CRYPTO'97, Springer-Verlag,LNCS 1294:165-179(1997).

[24] Y. Zheng, Signcryption and Its Applications in Efficient Public Key Solutions, Proc. of ISW'97-Information Security Workshop,LNCS 1396:291-312(1997).