This paper presents a novel proxy convertible multi-authenticated encryption (multi-AE) scheme and its variant with message linkages. The proposed scheme allows two or more original signers to cooperatively delegate their signing power to an authorized proxy signer, such that the proxy signer can generate a valid authenticated ciphertext on behalf of the original signing group and only a designated recipient is capable of decrypting the ciphertext and verifying its embedded proxy multi-signature. Its variant with message linkages further benefits the encryption of a large message by dividing it into many smaller message blocks. The proposed proxy convertible multi-AE scheme and its variant can simultaneously fulfill the security requirements of confidentiality and authenticity. Thus, they are applicable to those group-oriented confidential applications with proxy delegation, e.g., proxy on-line auction, proxy contract signing and so on. In case of a later dispute over repudiation, our proposed scheme also allows a designated recipient to convert the ciphertext into an original proxy multi-signature for public verification. In addition, the security of confidentiality against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) and that of unforgeability against existential forgery under adaptive chosen-message attacks (EF-CMA) are proved in the random oracle model.

**KEYWORDS:** proxy multi-signature, convertible, authenticated encryption, message linkage.

1. Introduction

With the rapid development of electronic commerce (eCommerce), the security of on-line transactions has received the great attention. Generally speaking, cryptographic techniques can be adopted to protect the communication content over the Internet. Public key encryptions [4] and digital signature schemes [5, 30] are two fundamental cryptographic mechanisms which primarily aim for providing confidentiality [9, 15] and authenticity [30], respectively. The digital signature scheme can further satisfy the requirement of non-repudiation [31] to prevent the signer’s dishonesty.

Some applications, however, like the contract signing, the savings withdrawal, on-line auctions and credit card transactions require all the above security re-
quirements simultaneously be achieved. A straightforward way would be sign-then-encrypt [36]. Yet, the approach is costly in terms of computation efforts and communication overheads. In some special circumstances, a proxy might be properly delegated to conduct these confidential transactions, e.g., a proxy auctions and the contract signing by an authorized proxy signer. Consider group-oriented applications such as the joint account owned by two or more individuals. To withdraw money from such account, all owners must cooperatively sign the withdrawal form which can only be verified by the bank teller. In case that account owners are unable to sign personally, they can delegate their signing power to a proxy signer who can legitimately conduct transactions on behalf of them. It thus can be seen that the design of efficient and provably secure cryptographic schemes fulfilling such requirements is crucial and benefits to the practical implementation.

1.2. Related Work

In 1996, Mambo et al. [25, 26] extended the concept of digital signature and introduced the notion of proxy signatures. 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. As to the proxy delegation, it can be categorized into four different kinds as follows:

(i). Full delegation [25, 26]: The proxy signer uses the private key which is the same as the one of the original signer so that all (proxy) signatures are signed with the same private key. Consequently, it is difficult for a verifier to identify the real signer from a given signature. That is to say, it cannot provide secure mechanisms to protect any one of the original and the proxy signers from being framed by the other.

(ii). Partial delegation [25, 26]: The proxy private key is further derived from the original signer’s one based on some cryptographic assumptions, e.g., the factorization and the discrete logarithm problems. It is infeasible to compute the original signer’s private key from the proxy one. Yet, it needs an additional revocation protocol as no information (e.g., the period of validity) is bonded to the delegation. Moreover, a malicious original signer can easily impersonate the proxy signer by deriving the corresponding proxy private key.

(iii). Delegation by warrant [28, 37]: A warrant which contains necessary proxy information, e.g., the period of validity and the identifiers of the original and the proxy signers, could be regarded as the delegation authorization. The warrant is then delivered to the proxy signer for convincing anyone. However, transmitting and verifying the certificate will incur extra computational and communicational costs.

(iv). Partial delegation with warrant [16]: This approach integrates the merits of partial delegation and delegation by warrant. It is also computationally infeasible for a proxy signer to derive the original signer’s private key from the proxy one. Besides, to certify the warrant and validate the signature can be simultaneously carried out within a single step, which helps reducing the computational and communicational costs.

Obviously, the fourth approach, partial delegation with warrant, is more flexible and secure as compared to the first three. Because of its efficiency and security compared with the others, we also adopt partial delegation with warrant to implement the proposed scheme. Up to the present, lots of variations of proxy signatures have been proposed [10, 12-14, 16, 23, 33, 34, 36].

In 1994, Horster et al. [8] proposed an authenticated encryption (AE) scheme which further provides digital signature schemes with the property of confidentiality and only the designated recipient can verify the signature instead of everyone. Since only the designated recipient has the ability to decrypt the ciphertext and verify the corresponding signature, there might be a potential drawback that the signer repudiates his signature. In such a circumstance, it is even difficult for an arbitrator to judge who is lying.

To deal with the case of a later dispute over repudiation, Araki et al. [1] proposed a convertible limited verifier signature scheme. However, the signature conversion of their scheme requires the assistance of the signer and incurs additional computation efforts, which is considered to be inefficient and unworkable if the signer is unwilling to cooperate with. Moreover, Zhang and Kim [48] also pointed out that their scheme can not withstand a universal forgery attack on an arbitrary chosen message.

In 2002, Wu and Hsu [40] proposed a convertible authenticated encryption (CAE) scheme, in which the
signature conversion is rather simple and can be solely done by the recipient without any computation effort or communication overhead. Huang and Chang [11] proposed an enhanced scheme in the next year. However, both the Wu-Hsu and the Huang-Chang schemes cannot fulfill the security requirement of confidentiality, i.e., the ciphertext is computationally distinguishable with respect to two candidate messages. To eliminate such a security weakness, Lv et al. [24] proposed a secure and practical solution. In 2005, Wu et al. [41] proposed generalized CAE schemes and adapted these schemes based on elliptic curves [17, 25] for facilitating gradually popular applications like smart cards [7], mobile phones and PDAs. Since then, lots of related works [6, 21, 35, 39, 43] have been proposed. In 2008, Chien [3] proposed a selectively CAE scheme allowing either the signer or the designated recipient to perform the signature conversion. In the next year, Lee et al. [19] addressed a CAE scheme based on the ElGamal cryptosystem. Considering the RSA cryptosystem, Wu and Lin [44] also presented a CAE scheme based on RSA in 2009. Nevertheless, these schemes are not suitable for the environment of multi-user setting. To fulfill group-oriented application requirements, Wu et al. [42] proposed a convertible multi-authenticated encryption (CMAE) scheme which enables a signing group composed of multiple signers to generate a valid authenticated ciphertext. In 2010, Tsai et al. [34] removed the necessity of using one-way hash functions. Based on Wu et al.’s scheme, Chang [2] addressed another variant with shared verification of multiple designated recipients. In 2012, Lu et al. [22] presented a provably CMAE scheme for generalized group communications. Later, Wu et al. [46] addressed a publicly verifiable PCAE scheme for confidential applications with proxy delegation. In 2014, Wu and Lin [45] proposed a proxy CAE scheme based on RSA. In 2015, a revocable CAE scheme [20] is also introduced. Yet, none of the above group-oriented CAE schemes can deal with the issue of proxy delegation. Although Lai and Singh [18] had proposed a similar scheme called ID-based multi-proxy multi-signcryption could solve the same problem, their scheme incurred time-consuming bilinear pairing operations and required extra key escrow mechanism.

1.3. Our Contribution

In this paper, we elaborate on the merits of CAE schemes and proxy multi-signature schemes to propose a novel proxy convertible multi-AE scheme and its variant with message linkages. The proposed scheme allows a delegated proxy signer to generate a valid authenticated ciphertext on behalf of the original signing group, such that only a designated recipient can recover the message and verify its embedded proxy multi-signature. When the case of a later dispute over repudiation occurs, a designated recipient can solely convert the authenticated ciphertext into a publicly verifiable proxy multi-signature without extra computation or communication cost. Besides, we further propose a variant with message linkages to benefit the encryption of a large message. We also prove that the proposed scheme achieves the security requirement of confidentiality against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) and that of unforgeability against existential forgery under adaptive chosen-message attacks (EF-CMA) in the random oracle model. As compared with related works, the proposed scheme not only provides better functionalities, but also has the provable security.

2. Preliminaries

In this section, we first describe the parties participating in the proposed scheme and define the composed algorithms.

2.1. Parties

Without loss of generality, there are \((n + 2)\) parties participating in a PCMAE scheme including a signing group (composed of \(n\) original signers), an authorized proxy signer and an intended recipient. All parties act as probabilistic polynomial-time Turing machines (PPTM). The original signers will deliver proxy credentials to the proxy signer. The latter is responsible for producing an authenticated ciphertext on behalf of the former. Yet, a dishonest proxy signer might repudiate his ciphertext. Finally, the intended recipient has the ability to decrypt the ciphertext and verify the embedded proxy multi-signature. A PCMAE scheme is said to be correct if the authorized proxy signer can generate a valid authenticated ciphertext and only the intended recipient is capable of decrypting it and verifying the proxy multi-signature.
2.2. Algorithms

The proposed PCMAE scheme has the following four algorithms:

- **Setup**: Initially, the Setup algorithm will generate public parameters utilized in the system. Let \( k \) be a security parameter. Taking as input \( 1^k \), the algorithm outputs the parameters \( \text{params} \).

- **Credential-gen (CG)**: The CG algorithm is used for producing the proxy credential for an authorized proxy signer. The input information of it is composed of the identity of proxy signer along with the private keys of all original signers. The resulted output is the corresponding proxy credentials.

- **Proxy-sign (PS)**: With the PS algorithm, an authorized proxy signer is able to generate a valid ciphertext on behalf of the original signing group. Thus, the input parameter includes a message \( m \), \( n \) proxy credentials, the intended recipient’s public key and the proxy signer’s private key. The corresponding output is an authenticated ciphertext \( \delta \).

- **Uncover-verify (UV)**: The UV algorithm is used for decrypting authenticated ciphertexts and checking the validity of embedded proxy multi-signature. It takes as input a ciphertext \( \delta \), a decryption key and all public keys of original and proxy signers. If the inputted ciphertext \( \delta \) is valid, it returns the signed message \( m \) and its converted proxy multi-signature \( \Omega \) which is publicly verifiable. Otherwise, an error symbol \( \bullet \) is returned instead.

3. The Proposed PCMAE Schemes

We give a concrete construction of our PCMAE scheme in this section. Also, a variant with message linkage for manipulating a large message will be presented. The used notations are stated as Table 1. Detailed steps for each algorithm are shown as follows:

### 3.1. Basic Construction

- **Setup**: Taking as input a security parameter \( k \), the system authority (SA) selects two large primes \( (p, q) \) and a generator \( g \) of order \( q \), where \( |q| = k \) and \( q \mid (p - 1) \). Let \( h_i \colon \{0, 1\}^k \times \mathbb{Z}_p \rightarrow \mathbb{Z}_q \), \( h_2 \colon \{0, 1\}^k \times \mathbb{Z}_p \times \mathbb{Z}_p \rightarrow \mathbb{Z}_q \) and \( h_3 \colon \mathbb{Z}_p \rightarrow \{0, 1\}^k \) be collision resistant hash functions. The system’s public parameters \( \text{params} = \{p, q, h, h_2, h_3\} \). Each user \( U_i \) chooses his private key \( x_i \in \mathbb{Z}_q \) and computes the public key as \( y_i = g^{x_i} \mod p \).

- **Credential-gen (CG)**: Let \( O = \{U_1, U_2, ..., U_n\} \) be the group of \( n \) original users delegating their signing power to the proxy signer \( U_p \). With the following steps, \( U_i \in O \) distributes the proxy share to \( U_p \).

**Step 1** \( U_i \in O \) first chooses \( t_i \in \mathbb{Z}_q \) to compute

\[
T_i = g^{t_i} \mod p, \tag{1}
\]

and then sends \( T_i \) to \( U_p \) and \( U_j \in O, j \neq i \).

**Step 2** Upon receiving all \( T_j \)’s, \( U_i \) computes

\[
T = \prod_{j=1}^{n} T_j \mod p \tag{2}
\]

\[
\sigma_i = t_i - x_i \cdot h_i(m_w, T) \mod q, \tag{3}
\]

where \( m_w \) is the warrant consisting of the identifier of the original and proxy signers, the delegation duration and so on. \( \sigma_i, m_w, T \) is then sent to \( U_p \).

**Step 3** Upon receiving \( (\sigma_i, m_w, T) \), \( U_p \) verifies

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Z_p )</td>
</tr>
<tr>
<td>( \mathbb{Z}_p^* )</td>
</tr>
<tr>
<td>( x \in \mathbb{Z}_p^* )</td>
</tr>
<tr>
<td>( x \leftarrow \mathbb{Z}_p^* )</td>
</tr>
<tr>
<td>( a \mod b )</td>
</tr>
<tr>
<td>( a \mid b )</td>
</tr>
<tr>
<td>(</td>
</tr>
<tr>
<td>( \sum_{r=1}^{n} \sum_{m=1}^{n} v_i )</td>
</tr>
<tr>
<td>( \prod_{r=1}^{n} \prod_{m=1}^{n} v_i )</td>
</tr>
<tr>
<td>( \otimes )</td>
</tr>
<tr>
<td>( \neg )</td>
</tr>
<tr>
<td>( \wedge )</td>
</tr>
<tr>
<td>( \vee )</td>
</tr>
<tr>
<td>( \forall )</td>
</tr>
<tr>
<td>( \Pr[E] )</td>
</tr>
</tbody>
</table>
\[ T_i = g^{\sigma_i} y_i^h(m_i, T) \mod p. \] (4)

If it does not hold, \((\sigma_i, m_i, T)\) is requested to be sent again.

We show that the verification of Eq. (4) works correctly. From the right-hand side of Eq. (4), we have

\[ g^{\sigma_i} y_i^h(m_i, T) = g^{\sigma_i - x_i h_i(x_i, T)} y_i^h(m_i, T) = g T_i \mod p \]

by Eq. (3)

which leads to the left-hand side of Eq. (4).

**Proxy-sign (PS):** For signing a message \(m \in \{0, 1\}^n\) on behalf of the original signing group \(O\), \(U_p\) chooses \(r \in \mathbb{Z}_p\) to compute

\[ R = g^r \mod p, \] (5)

\[ \sigma = \sum_{i=1}^{n} \sigma_i, \] (6)

\[ C = \prod_{i=1}^{n} y_i^h(m_i, T) \mod p, \] (7)

\[ K = y_v^\sigma \mod p, \] (8)

\[ S = r + (\sigma - x_v h_v(m, C, K, R)) \mod q, \] (9)

\[ Q = h_3(K) \oplus m, \] (10)

and then delivers the warrant \(m_w\) and the authenticated ciphertext \(\delta = (Q, S, R, T)\) to the designated recipient \(U_v\).

**Uncover-verify (UV):** Upon receiving \(\delta\), \(U_v\) first computes \(C\) as Eq. (7) and derives \(K\) as

\[ K = (TC^{-1})^x_v \mod p. \] (11)

He then recovers the message as

\[ m = Q \oplus h_3(K), \] (12)

and checks the redundancy embedded in \(m\). \(U_v\) can further verify the proxy multi-signature by checking if

\[ RT = g^S y_p^h(m, C, K, R) C \mod p. \] (13)

The correctness of Eqs. (12) and (13) can be easily confirmed. From the right-hand side of Eq. (12), we have

\[ Q \oplus h_3(K) = Q \oplus h_3((TC^{-1})^x_v \mod p) \] by Eq. (11)

\[ = Q \oplus h_3((g^\sigma)^{x_v} \mod p) \] by Eqs. (4), (6) and (7)

\[ = m \] by Eq. (8) and (10)

which leads to the left-hand side of Eq. (12).

If the authenticated ciphertext \((Q, S, R, T)\) is correctly generated, it will pass the test of Eq. (13). From the right-hand side of Eq. (13), we have

\[ g^S y_p^h(m, C, K, R) C = g^{r + \sigma - x_v h_v(m, C, K, R)} y_p^h(m, C, K, R) C \] by Eq. (9)

\[ = R g^\sigma C \] by Eq. (5)

\[ = RT \mod p \] by Eqs. (2), (4) and (6)

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

When a later dispute over repudiation occurs, \(U_v\) can reveal the converted proxy multi-signature \(\Omega = (S, R, T, K)\), the warrant \(m_w\) and the original message \(m\) to prove the proxy signer’s dishonesty without any additional cost. Thus, anyone can verify the converted proxy multi-signature with the assistance of Eqs. (7) and (13).

### 3.2. Variant with Message Linkages

Consider the practical implementation that the original message may be large. It therefore will cause the difficulty in encryption. In the subsection, we propose a variant with message linkages to benefit the encryption of a large message by dividing it into lots of small message blocks. The construction is similar as that in Section 3.1. We only describe the different parts as follows:

- **Proxy-sign (PS):** For signing a large message \(m\) on behalf of the original signing group \(O\), \(U_p\) first divides the message \(m\) into \(n\) pieces, i.e., \(m = m_1 || m_2 || \ldots || m_n\).
and recovers the original message $m_i$ by Eq. (10), and checks its validity with Eq. (12). From the left-hand side of Eq. (12), it leads to the left-hand side of Eq. (12).

Uncover-verify (UV): Upon receiving the ciphertext $(S, R, T, w_1, w_2, \ldots, w_n)$, the designated recipient $U_r$ can further verify the proxy multi-signature by checking Eq. (13). We then compute

$$m_i = w_i \cdot h_3(w_i - 1 \oplus h(K)) \mod p,$$

for $i = 1, 2, \ldots, n,$ (12)

and delivers the warrant $m_i$ along with $\delta = (S, R, T, w_1, w_2, \ldots, w_n)$ to the designated recipient $U_r$.

4. Security Proof and Comparison

In this section, we briefly review the security notions, state the security model and prove the security of our proposed scheme. Some comparisons with related schemes are also made.

4.1. Security Notions

Discrete Logarithm Problem; DLP

Let $p$ and $q$ be two large primes satisfying $q \mid p - 1$, and $g$ a generator of order $q$ over GF$(p)$. The discrete logarithm problem is, given an instance $(y, p, q, g)$, where $y = g^x \mod p$ for some $x \in \mathbb{Z}_q$, to derive $x$.

Discrete Logarithm (DL) 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 $A$, every positive polynomial $P(\cdot)$ and all sufficiently large $k$, the algorithm $A$ can solve the DLP with the advantage at most $1/P(k)$, i.e.,

$$\Pr[A(y, p, q, g) = \log_p g, (p, q, g) \leftarrow I_k, y \leftarrow \mathbb{Z}_q^*] \leq 1/P(k).$$

The probability is taken over the uniformly and independently chosen instance with a given security parameter $k$ and over the random choices of $A$.

Definition 1. The $(t, \varepsilon)$-DL assumption holds if there is no polynomial-time adversary that can solve the DLP in time at most $t$ and with the advantage $\varepsilon$.

Computational Diffie-Hellman Problem; CDHP

Let $p$ and $q$ be two large primes satisfying $q \mid p - 1$, and $g$ a generator of order $q$ over GF$(p)$. The computational Diffie-Hellman problem is, given an instance $(p, q, g, g^a, g^b)$ for some $a, b \in \mathbb{Z}_q$, to derive $g^{ab} \mod p$.

Computational Diffie-Hellman (CDH) 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 $A$, every positive polynomial $P(\cdot)$ and all sufficiently large $k$, the algorithm $A$ can solve the CDHP with the advantage at most $1/P(k)$, i.e.,

$$\Pr[A(p, q, g, g^a, g^b) = g^{ab}, (p, q, g) \leftarrow I_k, a, b \leftarrow \mathbb{Z}_q] \leq 1/P(k).$$

The probability is taken over the uniformly and independently chosen instance with a given security parameter $k$ and over the random choices of $A$.

Definition 2. The $(t, \varepsilon)$-CDH assumption holds if there is no polynomial-time adversary that can solve the CDHP in time at most $t$ and with the advantage $\varepsilon$.  

4.2. Security Model

The security requirements of the proposed PCMAE scheme and its variant are message confidentiality and unforgeability. The widely accepted notion for the security of message confidentiality comes from the definition of indistinguishability-based security, i.e., the adversary attempts to distinguish a target ci-
phertext with respect to two candidate messages. We define these notions as follows:

**Definition 3. (Confidentiality)** A PCMAE scheme is said to achieve the security requirement of confidentiality against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) if there is no probabilistic polynomial-time adversary \( \mathcal{A} \) with non-negligible advantage in the following game played with a challenger \( \mathcal{B} \):

**Setup:** The challenger \( \mathcal{B} \) first runs the Setup\( (1^k) \) algorithm and sends the system’s public parameters \( \text{params} \) to the adversary \( \mathcal{A} \).

**Phase 1:** The adversary \( \mathcal{A} \) can issue several kinds of queries adaptively, i.e., each query might be based on the result of previous queries:

- **Credential-gen (CG) queries:** \( \mathcal{A} \) makes a CG query with respect to the identity of target proxy signer. \( \mathcal{B} \) returns the corresponding proxy credentials.
- **Proxy-sign (PS) queries:** \( \mathcal{A} \) chooses a message \( m \) and then gives it to \( \mathcal{B} \) who will return a corresponding authenticated ciphertext \( \delta \) with the warrant \( m_w \).
- **Uncover-verify (UV) queries:** \( \mathcal{A} \) submits an authenticated ciphertext \( \delta \) along with the warrant \( m_w \) to \( \mathcal{B} \). If \( \delta \) is valid, \( \mathcal{B} \) returns the recovered message \( m \) and its converted proxy multi-signature \( \Omega \); else, the error symbol \( \bot \) is outputted as a result.

**Challenge:** The adversary \( \mathcal{A} \) produces two messages, \( m_1 \) and \( m_2 \), of the same length. The challenger \( \mathcal{B} \) flips a coin \( \lambda \leftarrow \{0,1\} \) and generates an authenticated ciphertext \( \delta^* \) for \( m_\lambda \). The ciphertext \( \delta^* \) 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 the UV query for the target ciphertext.

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

**Definition 4. (Unforgeability)** A PCMAE scheme is said to achieve the security requirement of unforgeability against existential forgery under adaptive chosen-message attacks (EF-CMA) if there is no probabilistic polynomial-time adversary \( \mathcal{A} \) with non-negligible advantage in the following game played with a challenger \( \mathcal{B} \):

**Setup:** \( \mathcal{B} \) first runs the Setup\( (1^k) \) algorithm and sends the system’s public parameters \( \text{params} \) to the adversary \( \mathcal{A} \).

**Phase 1:** \( \mathcal{A} \) issues the following kinds of queries adaptively:

**Forgery:** Finally, \( \mathcal{A} \) produces an authenticated ciphertext \( \delta^* \) which is not outputted by the PS query. The adversary \( \mathcal{A} \) wins if \( \delta^* \) is valid.

### 4.3. Security Proofs

We prove the security of our proposed scheme in the random oracle model as Theorems 1 and 2, respectively. The security proofs can be also applied to its variant with message linkages, since they have almost the same structure.

**Theorem 1. (Proof of Confidentiality)** The proposed scheme is \( \{t, q_{CG}, q_{PS}, q_{UV}, \varepsilon\} \)-secure against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) in the random oracle model if there is no probabilistic polynomial-time adversary that can \( (t', \varepsilon') \)-break the CDHP, where

\[
\varepsilon' \geq (q_{h_2} + q_{h_1})^{-1} (2\varepsilon - \frac{q_{UV}(q_{h_2} + q_{h_1} + 1)}{2^k}),
\]

\( t' \approx t + t\cdot (2q_{CG} + 4q_{PS} + 3q_{UV}). \)

Here \( t \) is the time for performing a modular exponentiation over a finite field.

**Proof:** Fig. 1 depicts the proof structure of this Theorem. Suppose that a probabilistic polynomial-time adversary \( \mathcal{A} \) can \( \{t, q_{h_1}, q_{h_2}, q_{CG}, q_{PS}, q_{UV}, \varepsilon\} \)-break the proposed scheme with non-negligible advantage \( \varepsilon \) under the adaptive chosen-ciphertext attack after running in time at most \( t \) and asking at most \( q_{h_1} \) random oracle (for \( i = 1 \) to \( 3 \)), \( q_{CG} \) CG, \( q_{PS} \) PS and \( q_{UV} \) UV queries. Then we can construct another algorithm \( \mathcal{B} \) that \( (t', \varepsilon') \)-breaks the CDHP by taking \( \mathcal{A} \) as a subroutine. Let all involved parties and parameters be defined the same as those in Section 3.1. The objective of \( \mathcal{B} \) is to obtain \( (g^{y'\cdot v} \mod p) \) by taking \( (p, q, g, y_p, y_i) \) as inputs. In this proof, \( \mathcal{B} \) simulates a challenger to \( \mathcal{A} \) in the following game.

**Setup:** The challenger \( \mathcal{B} \) runs the Setup\( (1^k) \) algorithm and sends the system’s public parameters \( \text{params} = \{p, q, g, y_p, y_i\} \) to the adversary \( \mathcal{A} \).

**Phase 1:** \( \mathcal{A} \) issues the following kinds of queries adaptively:
Figure 1
The proof structure of confidentiality in Theorem 1

\[ (p, q, g, y_p, y_v) \]

\[ h_1, h_2, h_3 \text{ oracles} \]

\[ \text{CG, PS, UV queries} \]

\[ g^{x_p + x_v} \mod p \]

- \( h_i \text{ oracle: When } \mathcal{A} \text{ makes an } h_i \text{ oracle of } (m_w, T), \mathcal{B} \text{ returns } \text{Sim}_h_i(m_w, T). \) The simulated random oracle \( \text{Sim}_h_i \) operates as Fig. 2. Note that the function \( \text{insert}(N, b) \) will insert the value \( b \) into the array \( N \).

- \( h_j \text{ oracle: When } \mathcal{A} \text{ makes an } h_j \text{ oracle of } (m, C, K, R), \mathcal{B} \text{ returns } \text{Sim}_h_j(m, C, K, R). \) The simulated random oracle \( \text{Sim}_h_j \) operates as Fig. 3.

- \( h_k \text{ oracle: When } \mathcal{A} \text{ makes an } h_k \text{ oracle of } K, \mathcal{B} \text{ returns } \text{Sim}_h_k(K). \) The simulated random oracle \( \text{Sim}_h_k \) operates as Fig. 4.

- \( \text{CG queries: When } \mathcal{A} \text{ makes a CG query, } \mathcal{B} \text{ chooses a proper } m_w \text{ and then returns } (m_w, \text{Sim}_{CG}(m_w)) \) as the result. The simulated CG oracle \( \text{Sim}_{CG} \) operates as Fig. 5. Note that the function \( \text{check}(N, b) \) will return a Boolean value depending on whether the value \( b \) is stored in the array \( N \).

- \( \text{PS queries: When } \mathcal{A} \text{ makes a PS query for some message } m, \mathcal{B} \text{ returns } \text{Sim}_{PS}(m) \) as the result. The simulated PS oracle \( \text{Sim}_{PS} \) operates as Fig. 6.

- \( \text{UV queries: When } \mathcal{A} \text{ makes a UV query for some authenticated ciphertext } \delta \text{ with the warrant } m_w, \mathcal{B} \text{ returns } \text{Sim}_{UV}(\delta, m_w) \) as the result. The simulated UV oracle \( \text{Sim}_{UV} \) operates as Fig. 7.

**Challenge:** \( \mathcal{A} \) generates two messages, \( m_0 \) and \( m_1 \), of the same length. The challenger \( \mathcal{B} \) flips a coin \( \lambda \leftarrow \{0, 1\} \) and produces an authenticated ciphertext \( \delta^* = (Q^*, S^*, R^*, T^*) \) for \( m_i \) by running the simulated Sim_Chal-
The algorithm of $\text{Sim}_\text{Challenge}$ operates as Fig. 8.

**Phase 2:** $A$ makes new queries as those stated in Phase 1 except the UV query for the target ciphertext $\delta$.

**Analysis of the game:** Consider the above simulations of CG and PS queries. One can see that the simulated proxy credentials $\sigma_i$'s and authenticated ciphertext $\delta$ are computationally indistinguishable from those generated by the real scheme. We refer the

**Figure 5**
Algorithm of the simulated CG oracle $\text{Sim}_\text{CG}$

```plaintext
oracle $\text{Sim}_\text{CG}(m_w)$
1: do
2: Choose $\sigma_i$'s, $v_1 \in R Z_q$;
3: $T_i = g^{\sigma_i} y_i^{\lambda_1} \mod p$;
4: $T = \prod_{i=1}^{n} T_i \mod p$;
5: while $(\text{check}(Q_{-h1}, (m_w, T)) = \text{true})$
6: $\text{insert}(Q_{-h1}, (m_w, T))$;
7: $\text{insert}(A_{-h1}, v_1)$;
     // define $h_1(m_w, T) = v_1$
8: return $(T, T_1, T_2, ..., T_n, \sigma_1, \sigma_2, ..., \sigma_n)$;
```

**Figure 6**
Algorithm of the simulated PS oracle $\text{Sim}_\text{PS}$

```plaintext
oracle $\text{Sim}_\text{PS}(m)$
1: Choose a proper $m_w$;
2: $(T, T_1, T_2, ..., T_n, \sigma_1, \sigma_2, ..., \sigma_n) \leftarrow \text{Sim}_\text{PS}(m_w)$;
3: Compute $v_1 \leftarrow \text{Sim}_\text{h1}(m_w, T)$;
     $\sigma = \sum_{i=1}^{n} \sigma_i$;
4: Compute $C = \prod_{i=1}^{n} y_i^{v_1} \mod p$; $K = y_v^{\sigma} \mod p$;
5: do
6: Choose $S, v_2 \in R Z_q$;
7: $R = g^S y_p^{v_2} CT^{-1} \mod p$;
8: while $(\text{check}(Q_{-h2}, (m, C, K, R)) = \text{true})$
9: $\text{insert}(Q_{-h2}, (m, C, K, R))$;
10: $\text{insert}(A_{-h2}, v_2)$; // define $h_2(m, C, K, R) = v_2$
11: $Q = \text{Sim}_\text{h3}(K) \oplus m$;
12: return $\delta = (Q, S, R, T)$ and $m_w$;
```

**Figure 7**
Algorithm of the simulated UV oracle $\text{Sim}_\text{UV}$

```plaintext
oracle $\text{Sim}_\text{UV}(\delta, m_w)$ // $\delta = (Q, S, R, T)$
1: $v_1 = \text{Sim}_\text{h1}(m_w, T); C = \prod_{i=1}^{n} y_i^{v_1} \mod p$
2: if $(\text{check}(Q_{-h2}, (*, C, *, R)) = \text{true})$ then
     // $h_2(*, C, *, R)$ has never been queried.
3: for $j = 0$ to $q_{h_2} - 1$
4: if $(Q_{-h2}[j][1] = C)$ and $(Q_{-h2}[j][3] = R)$ then
     $m = Q_{-h2}[j][0]$;
6: $K = Q_{-h2}[j][2]$;
7: $v_2 = A_{-h2}[j]$; exit for;
8: return $(m, R, S, T, K)$ and $m_w$;
9: end if
10: $v_3 = \text{Sim}_\text{h3}(K)$;
11: if $(m = Q \oplus v_3)$ and $(RT = g^S y_p^{v_2} C \mod p)$ then
12: return $(m, R, S, T, K)$ and $m_w$;
13: else
14: return $v_1$
15: end if
16: else // $h_2(*, C, *, R)$ has never been queried.
17: return $v_1$
18: end if
```

**Figure 8**
Algorithm of the simulated Sim\_Challenge

```plaintext
algorithm Sim\_Challenge($m_2$)
1: Choose a proper $m_2$; $v_3 \in R \{0, 1\}$ and $S^*, \sigma_1, v_2 \in R Z_q$;
2: $C = \prod_{i=1}^{n} y_i^{v_1} \mod p$; $T^* = y_p^{\sigma} C \mod p$;
3: $\text{insert}(Q_{-h1}, (m_2, T^*))$; $\text{insert}(A_{-h1}, v_1)$;
     // define $h_1(m_2, T^*) = v_1$
4: $R^* = g^S y_p^{v_2} CT^{-1} \mod p$;
5: $\text{insert}(Q_{-h2}, (m_2, C, \text{null}, R^*))$;
6: $\text{insert}(A_{-h2}, v_2)$;
     // Implicitly define $h_2(m_2, C, K^*, R^*) = v_2$, where $K^* = (y_v^{\sigma^2})^{\nu} \mod p$ and $B$ does not
     know it.
7: $Q^* = v_3 \oplus m$; // Implicitly define $h_3(K^*) = v_3$
8: return $\delta = (Q^*, S^*, R^*, T^*)$ and $m_2$;
```
simulations of CG and PS queries to be perfect. Then we evaluate the simulation of UV queries. From the algorithms of _Sim_UV_, we find out that it is possible for an UV query of some valid δ = (Q, S, R, T) to return the error symbol Ψ on condition that _A_ has the ability to produce δ without asking the corresponding _h_2(m_j, C, K, R) or _h_2(K) random oracles in advance. Let UV_ERR be the event that an UV query returns the error symbol Ψ for some valid δ during the entire game, AC-V an event that the authenticated ciphertext δ of a UV query made by _A_ is valid. QH1 and QH2 separately denote the events that _A_ has ever asked _h_2(m_j, C, K, R) and _h_2(K) random oracles beforehand. Then we can express the error probability of any UV query as

\[
Pr[AC-V | (¬QH3 ∨ ¬QH2)]
\]

\[
≤ Pr[AC-V | ¬QH3] + Pr[AC-V ∧ QH3 | ¬QH2]
\]

\[
= Pr[AC-V ∧ QH2 | ¬QH3] + Pr[AC-V ∧ ¬QH2 | ¬QH3] + Pr[AC-V ∧ QH3 | ¬QH2]
\]

\[
≤ \frac{q_{h2} + 1}{2^k} + \frac{q_{h1}}{2^k} + \frac{q_{h1}}{2^k}
\]

\[
= \frac{q_{h2} + q_{h1} + 1}{2^k}.
\]

Since _A_ can make at most _q_UV_ UV queries, we can further express the probability of UV_ERR as

\[
Pr[UV\_ERR] ≤ \frac{q_{UV}(q_{h2} + q_{h1} + 1)}{2^k}.
\]

(14)

Additionally, in the challenge phase, _B_ has returned a simulated authenticated ciphertext δ* = (Q', S', R', T') where T' = y^c mod p, which implies the shared secret K* is implicitly defined as (y^c) mod p. Let GP be the event that the entire simulation game does not abort. Obviously, if the adversary _A_ never asks _h_2(m_j, C, K*, R*) or _h_2(K*) random oracles in Phase 2, the entire simulation game could be normally terminated. We denote the two events that _A_ does make an _h_2(m_j, C, K*, R*) and _h_2(K*) query in Phase 2 by QH star 1 and QH star 2. When the entire simulation game does not abort, it can be seen _A_ gains no advantage in guessing λ due to the randomness of the output of the random oracle, i.e.,

\[
Pr[\lambda' = \lambda | GP] = 1/2.
\]

(15)

Rewriting the expression of Pr[λ' = λ], we have

\[
Pr[\lambda' = \lambda] = Pr[\lambda' = \lambda | GP] Pr[GP]
\]

\[
+ Pr[\lambda' = \lambda | ¬GP] Pr[¬GP]
\]

\[
≤ (1/2)Pr[GP] + Pr[¬GP]
\]

by Eq. (15).

\[
= (1/2)(1 - Pr[¬GP]) + Pr[¬GP]
\]

\[
= (1/2) + (1/2)Pr[¬GP].
\]

(16)

On the other hand, we can also derive that

\[
Pr[\lambda' = \lambda] ≥ Pr[\lambda' = \lambda | GP] Pr[GP]
\]

\[
= (1/2)(1 - Pr[¬GP])
\]

\[
= (1/2) - (1/2)Pr[¬GP].
\]

(17)

With inequalities (16) and (17), we know that

\[
| Pr[\lambda' = \lambda] - 1/2 | ≤ (1/2)Pr[¬GP].
\]

(18)

Recall that in Definition 3, _A_’s advantage is defined as Adv(λ) = | Pr[λ' = λ] - 1/2 |. By assumption, _A_ has non-negligible probability ε to break the proposed scheme. We therefore have

\[
ε = | Pr[λ' = λ] - 1/2 |
\]

\[
≤ (1/2)Pr[¬GP]
\]

\[
≤ (1/2)(Pr[QH2* ∨ QH3* ∨ UV\_ERR]) + Pr[QH*] + Pr[QH3*] + Pr[UV\_ERR])
\]

by Eq. (18).

Combining Eq. (14) and rewriting the above inequality, we get

\[
(Pr[QH2*] + Pr[QH3*]) ≥ 2ε - Pr[UV\_ERR]
\]

\[
≥ 2ε - \frac{q_{UV}(q_{h2} + q_{h1} + 1)}{2^k}.
\]

If the event (QH2* ∨ QH3*) happens, we claim that K* = (y^c) mod p will be stored in some entry of the Q_h_j or the Q_h_k array. Consequently, _B_ has non-negligible probability

\[
ε' ≥ (q_{h2} + q_{h3})^{-1}(2ε - \frac{q_{UV}(q_{h2} + q_{h1} + 1)}{2^k})
\]

to output K*σ^-1 = g^x_r^x_v and solve the CDHP. The computational time required for _B_ is t' ≈ t + t_h(2q_{ca} + 4q_{ps} + 3q_{uv}).

Q.E.D.
In 2000, Pointcheval and Stern introduced the Forking lemma [29] to prove the security for generic digital signature schemes in the random oracle model. If we apply their techniques to prove our scheme, we can also obtain the generic result as follows.

**The Forking Lemma** In the random oracle model, let \((\mathcal{G}, \Sigma, \mathcal{V})\) be a generic signature scheme and \(\mathcal{A}\) a probabilistic polynomial-time Turing machine whose input only consists of public data. We denote respectively by \(N_1\) and \(N_2\) the number of queries that \(\mathcal{A}\) can ask to the random oracle and the number of queries that \(\mathcal{A}\) can ask to the signer. Assume that, within a time bound \(T\), \(\mathcal{A}\) produces, with probability \(\varepsilon \geq 10 (N_1 + 1)(N_2 + N_3)/2^n\), a valid signature \((m, \sigma, h, \Sigma)\) where \(\sigma_i = (m_{\sigma_i}, R, T, K), h = (h_{\Sigma}(m, C, K, R), h_{\Sigma}(m, T))\) and \(\sigma_i = v\). If the triples \((\sigma, h, \Sigma)\) can be simulated without knowing the private key with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from \(\mathcal{A}\) replacing interaction with the signer by simulation and produces two valid signatures \((m, \sigma, h, \Sigma)\) and \((m, \sigma, h', \Sigma')\) such that \(h_{\Sigma}(m, C, K, R) \neq h'_{\Sigma}(m, C, K, R)\) in the expected time \(T' \leq 1200686 T/\varepsilon\).

More concretely, in our scheme, we can first obtain two equations below:
\[
RT = g^n y_p h_{\Sigma}(m, C, K, R) \bmod p, \\
RT = g^n y_p h'_{\Sigma}(m, C, K, R) \bmod p.
\]

By combining the above two equalities, we can further derive the private key \(x_p\) as
\[
x_p = (S - S')/(h'_{\Sigma}(m, C, K, R) - h_{\Sigma}(m, C, K, R)).
\]

Still, to give a tight reduction from the hardness of DLP to our proposed scheme, we present another more detailed security proof and the advantage analysis as Theorem 2.

**Theorem 2. (Proof of Unforgeability)** The proposed scheme is \((t, q_h, q_y, q_p, q_v, q_{cv}, q_{ps}, \varepsilon)\)-secure against existential forgery under adaptive chosen-message attacks (EF-CMA) in the random oracle model if there is no probabilistic polynomial-time adversary that can \((t', \varepsilon')\)-break the DLP, where
\[
\begin{align*}
\varepsilon' &\geq A^{-1}(\varepsilon - 2^{-2^n q_{\text{ps}}}), \\
t' &\leq t + t_0(4q_{cv} + 8q_{ps}).
\end{align*}
\]

Here \(t_0\) is the time for performing a modular exponentiation over a finite field.

**Proof:** Fig. 9 depicts the proof structure of this theorem. Suppose that \(\mathcal{A}\) is a probabilistic polynomial-time adversary \(\mathcal{A}\) can \((t, q_h, q_y, q_p, q_v, q_{cv}, q_{ps}, \varepsilon)\)-break the proposed scheme with non-negligible advantage \(\varepsilon\) under the adaptive chosen-message attack after running in time at most \(t\) and asking at most \(q_h, h_1, q_y, h_1, q_p, h_2, q_{cv}\) and \(q_{ps}\) PS queries. Then we can construct another algorithm \(\mathcal{B}\) that \((t', \varepsilon')\)-breaks the DLP by taking \(\mathcal{A}\) as a subroutine. Let all involved parties and notations be defined the same as those in Section 3.1, \(h_1\) a collision resistant hash function and \(\{h_1, h_2\}\) random oracles. The objective of \(\mathcal{B}\) is to obtain \(x_p = \log_g y_p\) by taking \((p, q, g, y_p)\) as inputs. In this proof, \(\mathcal{B}\) simulates a challenger to \(\mathcal{A}\) in the following game.

**Setup:** The challenger \(\mathcal{B}\) runs the Setup\((1^n)\) algorithm to obtain the system’s public parameters \(\text{params} = \{p, q, g\}\) and comes up with a random tape composed of a long sequence of random bits. Then \(\mathcal{B}\) simulates two runs of the proposed scheme to the adversary \(\mathcal{A}\) on input \(\text{params}, y_p, y_v = g^{e\alpha} \bmod p\), where \(\alpha \in R Z_q^*_r\) and the random tape.

**Phase 1:** \(\mathcal{A}\) adaptively asks \(h_1, h_2\) random oracles, \(CG\) and \(PS\) queries as those defined in Theorem 1.

**Analysis of the Game:** According to the analyses of Theorem 1, the simulations of \(CG\) and \(PS\) queries are perfect. Namely, the adversary \(\mathcal{A}\) can not distinguish...
whether he is playing in either a simulation or a real scheme. Let AC-V be the event that \( A \) forges a valid authenticated ciphertext \( \delta = (Q, S, R, T) \) for his arbitrarily chosen message \( m \). Since \( A \) has non-negligible probability \( \varepsilon \) to break the proposed scheme under the adaptive chosen-message attack by the initial assumption, we know that

\[
\Pr[AC-V] = \varepsilon.
\]

Now we further consider the situation where \( A \) is able to output a valid \( \delta \) without asking \( h_1 \) and \( h_2 \) random oracles in advance. Let NR be the event that \( A \) guesses correct output values of \( h_1(m, T) \) and \( h_2(m, C, K, R) \) without asking the random oracles, i.e., \( \Pr[NR] \leq 2^{-2k} \). Then, we can express the probability that \( A \) outputs a valid forgery \( \delta = (Q, S, R, T) \) after asking the corresponding random oracles as

\[
\Pr[AC-V \land \neg NR] \geq (\varepsilon - 2^{-2k}).
\]

With the initially selected private key \( \alpha, \beta \) can recovers \( m \) and obtain the multi-proxy signature \((S, R, T, K)\) along with \( m_w \).

Then \( \beta \) launches the second simulation. He again runs \( A \) on input \( \text{params}, y, y_v, y_s = g^v \mod p \) where \( \alpha \in_R Z_q \) and the same random tape. Since the adversary \( A \) is given the same sequence of random bits, we can anticipate that the \( i \)-th random query \( A \) asks will always be the same as the one in the first simulation. In the second simulation, \( \beta \) returns identical results as those he responds in the first time until \( A \) makes the \( h_1(m, C, K, R) \) query. At this time, \( \beta \) directly gives another answer \( v_s^* \in_R Z_q \) rather than original \( v_s \). Meanwhile, \( A \) is then supplied with a different random tape which also consists of a long sequence of random bits. From the statement of “Forking lemma”, we can learn that when \( A \) finally makes another valid forgery \( \delta^* = (Q^*, S^*, R, T^*) \) where \( h_1(m, C, K, R) \neq h_1^*(m, C, K, R) \), \( \beta \) could solve the DLP with non-negligible probability. To analyze \( \beta \)'s success probability, we use the “Splitting lemma” [29] described below:

Let \( X \) and \( Y \) be the sets of possible sequences of random bits and random function values provided to \( A \) before and after the \( h_1(m, C, K, R) \) query is issued, respectively. It follows that on inputting a random value \( (x \mid y) \) for any \( x \in X \) and \( y \in Y \), \( A \) returns a valid forgery with the non-negligible probability \( \varepsilon \), i.e.,

\[
\Pr_{x \in X, y \in Y}[AC-V] = \varepsilon.
\]

By the “Splitting lemma”, there exists a subset \( D \in X \) such that

(a) \( \Pr[x \in D] = |D| \cdot |X|^{-1} \geq 2^{-1} \varepsilon \).

(b) \( \forall x \in D, \Pr_{y \in Y}[AC-V] \geq 2^{-1} \varepsilon \).

If we let \( \rho \in D \) and \( y^* \in Y \) separately be the supplied sequences of random bits and random function values before and after \( A \) makes the \( h_1(m, C, K, R) \) query, \( A \) is able to make a valid forgery in the second simulation with the probability of at least \((2^{-1}\varepsilon)^2 = 4^{-1}\varepsilon^2\), i.e.,

\[
\Pr_{\rho \in D, y^* \in Y}[AC-V] \geq 4^{-1}\varepsilon^2.
\]

Since we have known that \( A \) eventually returns another valid \( \delta^* = (Q^*, S^*, R, T^*) \) with \( h_2(m, C, K, R) \neq h_2^*(m, C, K, R) \) is \( q \)\(^{-1}\), the probability of \( \delta \) to solve the DLP in the second simulation can be represented as

\[
\varepsilon^* \geq (\varepsilon - 2^{-2k})(4^{-1}(\varepsilon - 2^{-2k})^2)(q^{-1})^2 \geq (\varepsilon - 2^{-2k})(4^{-1}(\varepsilon - 2^{-2k})^2)(q^{-1}).
\]

Moreover, the computational time required for \( \beta \) in one simulation is

\[
t \approx t + t_A(4q_C + 8q_{GD}).
\]

Q.E.D.

According to Theorem 2, the proposed scheme is secure against existential forgery attacks. That is, the proxy private key can not be forged and the delegated proxy signer can not repudiate having generated his authenticated ciphertext. Hence, we obtain the following corollary.

**Corollary 1.** The proposed scheme satisfies the security requirement of non-repudiation.

### 4.4. Comparisons

We compare the proposed scheme with some related works including Lv et al.’s (LWK for short) [24], Tso et al.’ (TOO for short) [35], Araki et al.’ (AUI for short) [1], the Wu-Hsu (WH for short) [40], Wu et al.’s (WHT for short) [42], Chang’s (Cha for short) [2], Tsai’s (Tsa for short) [34] and the Lin-Yeh (LY for short) [22] schemes in terms of functionalities and security proofs. Detailed comparisons are demonstrated as Table 2. Since WHT and Cha schemes also have provable security, we further compare our work with them in terms of computational efforts which is evaluated by the number of required modular exponentiation operations. The performance comparison is demonstrated as Table 3. From these tables, it can be seen that the proposed scheme provides not only better functionalities, but also lower computational costs.
5. Conclusions

In this paper, we have proposed a novel PCMAE scheme to solve the group-oriented delegation problem for confidential transactions. The proposed scheme allows the proxy signer to produce an authenticated ciphertext on behalf of the original signing group and only the designated recipient is capable of recovering the message and verifying its proxy multi-signature for guaranteeing the confidentiality. Its variant with message linkages further benefits the transmission of a large message by dividing it into many smaller message blocks. It is not necessary to establish a session key in advance between a proxy signer and a designated recipient. Without revealing the private key, a designated recipient can independently convert the authenticated ciphertext into an ordinary proxy multi-signature for the public arbitration in case of a later repudiation. Since the converted proxy multi-signature is obtained during the message recovery and signature verification process, the signature conversion process requires no extra computation efforts and communication overheads. In addition, we also proved that the proposed scheme achieves the security requirement of confidentiality against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) and that of unforgeability against existential forgery under adaptive chosen-message attacks (EF-CMA) in the random oracle model. As compared with related works, ours not only provides better functionalities, but also has provable security.

Acknowledgements

The author would like to thank anonymous referees for their valuable suggestions. This work was supported in part by the Ministry of Science and Technology of Republic of China under the contract number MOST 106-2221-E-019-008.
References


47. Xue, Q., Cao, Z. A Nonrepudiable Multi-Proxy Multisignature Scheme. Proceedings of 1st Joint Workshop on
This paper presents a novel proxy convertible multi-authenticated encryption (multi-AE) scheme and its variant with message linkages. The proposed scheme allows two or more original signers to cooperatively delegate their signing power to an authorized proxy signer, such that the proxy signer can generate a valid authenticated ciphertext on behalf of the original signing group and only a designated recipient is capable of decrypting the ciphertext and verifying its embedded proxy multi-signature. Its variant with message linkages further benefits the encryption of a large message by dividing it into many smaller message blocks. The proposed proxy convertible multi-AE scheme and its variant can simultaneously fulfill the security requirements of confidentiality and authenticity. Thus, they are applicable to those group-oriented confidential applications with proxy delegation, e.g., proxy on-line auction, proxy contract signing and so on. In case of a later dispute over repudiation, our proposed scheme also allows a designated recipient to convert the ciphertext into an original proxy multi-signature for public verification. In addition, the security of confidentiality against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) and that of unforgeability against existential forgery under adaptive chosen-message attacks (EF-CMA) are proved in the random oracle model.