ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2014, T. 43, Nr. 3

Group-Oriented Data Access Structure Using Threshold-CAE Scheme and Its Extension

Han-Yu Lin

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan
e-mail: lin.hanyu@msa.hinet.net

crossref http://dx.doi.org/10.5755/j01.itc.43.3. 5708

Abstract. Conventional authenticated encryption (AE) schemes put emphasis on the single-user setting, which only allow one signer to produce an authenticated ciphertext such that merely the designated recipient is capable of recovering the message and verifying its corresponding signature. In the multi-user environments, e.g., organizational operations, several senior managers might cooperatively sign a confidential business contract according to the organizational signing policies. To fulfill such application requirements, in this paper, we propose a secure (t, n) threshold convertible authenticated encryption (TCAE) scheme and its variant with message linkages for the multi-user environment. In our proposed scheme, any t or more signers can cooperatively generate a valid authenticated ciphertext while less than or equal to t–1 cannot. In case of a later dispute over repudiation, the designated recipient can solely convert the authenticated ciphertext into an ordinary multi-signature without extra computational efforts for protecting his benefits. Moreover, the security requirement of confidentiality against adaptive chosen-ciphertext attacks (IND-CCA2) and that of unforgeability against existential forgery on adaptive chosen-message attacks (EF-CMA) are proved in the random oracle model. Compared with related works, our scheme provides not only better functionalities, but also lower computational costs.

Keywords: data access structure; authenticated encryption; threshold; formal proof; multi-user.

1. Introduction

The public key cryptosystem was first introduced by Diffie and Hellman [1] in 1976. Based on the intractability of solving the discrete logarithm problem (DLP) [1, 2], the public key system equips each user with a private key and the corresponding public key which is accessible to anyone. It is computationally infeasible for any malicious adversary to derive the private key from its known public one. The public key encryption and digital signature schemes [3-5] are two vital mechanisms of public key systems. When communicating over an insecure channel like the Internet, a sender can deliver a message encrypted with the receiver's public key to the destination such that only the intended receiver can decrypt the ciphertext with his own private key. It thus can be seen that the public key encryption fulfills the security requirement of confidentiality [6]. As to further achieving the property of authenticity [7], an authenticated encryption (AE) scheme introduced by Horster *et al*. [8] is applicable. Such schemes enable a signer to generate an authenticated ciphertext such that only the designated recipient has the ability to recover the message and verify its corresponding signature. It is not necessary to establish a secret channel between the signer and the designated recipient in advance. Yet, a later dispute that the signer repudiates his signatures might occur. To eliminate the drawback, in 1999, Araki *et al*. [9] proposed a convertible limited verifier signature scheme which provides the signature conversion mechanism to deal with the dispute. However, some extra computational cost will be incurred during the conversion. In 2002, Wu and Hsu [10] proposed a convertible authenticated encryption (CAE) scheme in which the signature conversion process is rather simple and can be solely done by the designated recipient without any additional communicational and computational cost. That is to say, CAE schemes further satisfy the requirement of non-repudiation [11]. In 2005, Chen and Jan [12] proposed CAE schemes using self-certified public key system [13]. Peng *et al.* [14] addressed a publicly verifiable authenticated encryption scheme with message linkages for transmitting a large message. Later, Lv *et al.* [15] further proposed a more secure and practical CAE scheme to improve the Wu-Hsu scheme. To orient this research topic, Hwang and Liu [16] have given detailed overview and analyses in relation to the key issues of AE/CAE schemes.

With the diversified development of E-Commerce, group-oriented applications play an important role in the modern society. In the multi-user environments, e.g., organizational operations, several senior managers might cooperatively sign a confidential business contract. In such a case, traditional cryptographic schemes focusing on the single-user setting are not applicable here. Although threshold signature schemes [17] permit a subset of signers to produce a valid signature on behalf of the entire signing group, they are not suitable for the situation where confidentiality is regarded as a crucial property. To fulfill the above group-oriented application requirement, in 2008, Wu *et al.* [18] proposed a convertible multi-authenticated encryption scheme which enables a signing group composed of multiple signers to generate a valid authenticated ciphertext. In 2009, Tsai [19] improved Wu *et al.*'s scheme by reducing the computational costs and removing the necessity of message redundancy. However, we find out that Tsai's scheme still cannot satisfy the security requirement of indistinguishability since anyone can easily identify the encrypted message from two candidate messages for a given ciphertext. Based on Wu *et al.*'s scheme, Chang [20] also presented another variant with shared verification of multiple designated recipients. Lin and Yeh [21] further proposed a threshold convertible authenticated encryption (TCAE) scheme allowing any t or more signers to cooperatively generate a valid authenticated ciphertext on behalf of the original signing group. Nevertheless, the computational costs of the Lin-Yeh scheme are rather high and no formal security proofs are given. Considering the key-compromise problem in 2011, Hsu and Lin [22] proposed an identity-based key-insulated convertible multi-authenticated encryption scheme.

In this paper, we propose a secure (t, n)-TCAE scheme and its variant with message linkages. The variant with message linkages is especially suitable for the transmission of a large message over the public network. When the signing group repudiates having generated their authenticated ciphertext, the designated recipient can convert the authenticated ciphertext into an ordinary multi-signature for public verification without neither extra computational costs nor the cooperation of the signers. Besides, the security requirement of confidentiality against adaptive chosen-ciphertext attacks (IND-CCA2) and that of unforgeability against existential forgery on adaptive chosen-message attacks (EF-CMA) are proved in the random oracle model. To the best of our knowledge, the proposed scheme is the first provably secure (t, n)-TCAE scheme based on the computational Diffie-Hellman problem (CDHP). Compared with previous works, ours provides not only better functionalities, but also lower computational costs.

The rest of this paper is organized as follows. Section 2 states some preliminaries. The formal model of our proposed scheme is defined in Section 3. We introduce the proposed (t, n)-TACE scheme and its variant with message linkages in Section 4. Some comparisons and the security proofs are detailed in Section 5. Finally, a conclusion is made in Section 6.

2. Preliminaries

In this section, we briefly review some security notions and the concept of random oracle model.

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 $y = g^x \bmod p$ for some $x \in Z_q$, 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 $\mathcal{A}$, every positive polynomial $P(\cdot)$ and all sufficiently large k, the algorithm $\mathcal{A}$ can solve the DLP with an advantage at most $1/P(k)$, i.e.,

$$\Pr[\mathcal{A}(y, p, q, g) = \mathrm{Log}_{p,q,g}(y),$$

$$(p, q, g) \leftarrow I_k, y \leftarrow Z_p^*] \leq 1/P(k).$$

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

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

Computational Diffie-Hellman Problem; CDHP

Let p and q be two large primes satisfying that $q|p-1$ 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 $a, b \in Z_q$, to derive $g^{ab} \bmod p$.

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 $\mathcal{A}$, every positive polynomial $P(\cdot)$ and all sufficiently large k, the algorithm $\mathcal{A}$ can solve the CDHP with an advantage at most $1/P(k)$, i.e.,

$$\Pr[\mathcal{A}(p, q, g, g^a, g^b) = g^{ab},$$

$$(p, q, g) \leftarrow I_k, a, b \leftarrow Z_q] \leq 1/P(k).$$

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

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

Random Oracle Model

In the random oracle model, a cryptographic hash function is simulated as a random oracle which must be queried in order to obtain the corresponding output with respect to a given input. That is to say, an adversary can query a hash oracle for his chosen input and a challenger will return the response. An adaptive chosen-message attacker (CMA) is also allowed to query the signature for his chosen messages adaptively while an adaptive chosen-ciphertext attacker (CCA2) is further given the ability to query the plaintext for his chosen ciphertexts for several times. A signature scheme is said to be CMA-secure in the random oracle model if there is no polynomial-time adversary that can forge a valid signature with non-negligible advantage. Similarly, an encryption mechanism is said to be CCA2-secure in the random oracle model if there is no polynomial-time adversary that can decrypt the target challenge with non-negligible advantage.

3. Formal Model of Our Proposed Scheme

This section defines the formal model of our proposed (t, n)-TCAE scheme.

3.1. Involved Parties

A (t, n)-TCAE scheme has two kinds of involved parties: a group of n signers and a designated recipient. Each one is a probabilistic polynomial-time Turing machine (PPTM). Any t or more signers can cooperatively produce a valid authenticated ciphertext on behalf of the group while less than or equal to $t - 1$ cannot. Finally, the designated recipient decrypts the ciphertext and verifies the multi-signature.

3.2. Composed Algorithms

The proposed (t, n)-TCAE scheme consists of the following algorithms:

- **Setup:** Taking as input 1^k where k is a security parameter, the algorithm generates the system's public parameters *params*.
- **Authenticated-Ciphertext-Generation (ACG):**

  The ACG algorithm takes as input the system parameters *params*, a message m, the public key of designated recipient and the private keys of at least t signers. It generates the resulted authenticated ciphertext δ.
- **Signature-Recovery-and-Verification (SRV):**

  The SRV algorithm takes as input the system parameters *params*, an authenticated ciphertext δ, the private key of designated recipient and the public key of the signing group. It outputs the message m and its converted multi-signature Ω if the authenticated ciphertext δ is valid. Otherwise, the symbol ¶ is returned as a result.

4. The Proposed Scheme

In this section, we introduce the proposed scheme along with its variant over a finite field and then demonstrate its correctness. One realistic application for our proposed scheme is business contract signing. Suppose that a board of directors for some company consists of n persons. According to the regulation, a valid contract must be signed by at least t directors where $t \leq n$. Since the content of this business contract is confidential, only the lawyer of corresponding company is able to verify it. Based on the roles of the above example, the board of n directors could be regarded as the original signing group, t directors who have signed the contract are the actual signing subgroup and the lawyer is viewed as the designed verifier in the following construction.

In the proposed scheme, there are three main phases including Setup, Authenticated-Ciphertext-Generation (ACG) and Signature-Recovery-and-Verification (SRV). In Setup phase, a system authority is responsible for generating necessary system parameters along with each user's key pair. In ACG phase, a subgroup of t signers will cooperatively generate a valid authenticated ciphertext with the assistance of a clerk. Finally, in SRV phase, a designated verifier can decrypt the ciphertext and verify the corresponding multi-signature. If necessary, the designated verifier has the right to reveal a converted multi-signature for public verification.

4.1. Construction

- **Setup:** Taking as input 1^k, the system authority (SA) selects a $t - 1$ degree polynomial $f(\varpi) = d_0 + d_1\varpi + \ldots + d_{t-1}\varpi^{t-1}$ for all d_i's $\in Z_q$, two large primes p and q where $|q| = k$ and $q \mid (p - 1)$, and g a generator of order q. Let h_1: $\{0, 1\}^k \times Z_p^* \rightarrow Z_q$, h_2: $\{0, 1\}^k \rightarrow Z_p^*$, h_3: $Z_p^* \rightarrow \{0, 1\}^k$ and h_4: $Z_p^* \rightarrow Z_q$ be collision resistant hash functions. The system publishes the public parameters $params = \{p, q, g, h_1, h_2, h_3\}$ and derives each user U_i's private key $x_i = f(i)$. The corresponding public key is computed as $y_i = g^{x_i} \bmod p$.
- **Authenticated-Ciphertext-Generation (ACG):**

  Without loss of generality, let $O = \{U_1, U_2, \ldots, U_n\}$ be the signing group, $SO = \{U_1, U_2, \ldots, U_t\}$ the subgroup composed of t signers who cooperatively generate a valid authenticated ciphertext on behalf of O, and U_{ck} a semi-trusted clerk who is responsible for verifying individuals' signatures and combining them into the corresponding authenticated ciphertext. A semi-trusted third party is said to be honest but curious, i.e., he will not perform anything that deviates from the predefined procedures, but he might attempt to learn any secret information from observed messages. The private key of O is d_0 and the corresponding public key is $y_D = g^{d_0} \bmod p$. For signing the message $m \in_R \{0, 1\}^k$, each $U_i \in SO$ first chooses $r_i \in_R Z_q$ to compute the Lagrange coefficient [23] c_i as

  $$c_i = \prod_{U_j \in SO \setminus \{U_i\}} j/(j-i) \bmod q, \qquad (1)$$

$$e_i = c_i \cdot x_i \bmod q, \tag{2}$$

$$R_i = g^{r_i} \bmod p, \tag{3}$$

and then sends R_i to $U_j \in SO \setminus \{U_i\}$ and U_{ck}. Upon receiving all R_j's, $U_i \in SO$ computes

$$R = \prod_{j=1}^{n} R_j h_2(m) \bmod p\,, \tag{4}$$

$$s_i = r_i - e_i h_1(m, R) \bmod q\,. \tag{5}$$

s_i is then delivered to the clerk U_{ck} via a secure channel. After receiving all s_i's, U_{ck} verifies whether

$$R_i = g^{s_i} y_i^{c_i h_1(m,R)} \bmod p\,. \tag{6}$$

If it does not hold, s_i is requested to be sent again; else, U_{ck} chooses $\theta \in_R Z_q$ to compute

$$s = \sum_{U_i \in SO} s_i \bmod q\,, \tag{7}$$

$$K = y_v^{(s + \theta)} \bmod p, \tag{8}$$

$$T = g^{(s + \theta)} \bmod p, \tag{9}$$

$$Q_1 = sh_4(K), \tag{10}$$

$$Q_2 = h_3(K) \oplus m. \tag{11}$$

The authenticated ciphertext $\delta = (Q_1, Q_2, R, T)$ is then delivered to the designated recipient U_v.

- **Signature-Recovery-and-Verification (SRV):**

  Upon receiving (Q_1, Q_2, R, T), U_v first computes

$$K = (T)^{x_v} \bmod p, \tag{12}$$

$$s = h_4(K)^{-1} Q_1, \tag{13}$$

$$m = Q_2 \oplus h_3(K), \tag{14}$$

  and then checks the redundancy embedded in m. U_v can further verify the multi-signature by checking

$$R = g^s y_D^{h_1(m,R)} h_2(m) \pmod{p}. \tag{15}$$

When the case of a later dispute over repudiation occurs, U_v can announce the converted multi-signature $\Omega = (R, s)$ and the message m to convince the third party of the signing group's dishonesty without any additional computational effort or communicational overhead. Therefore, with the assistance of Eq. (15), anyone can verify the converted multi-signature.

4.2. Correctness

The correctness proof includes two parts: correct recovery of the message and effective verification of the multi-signature. To recover the message, the designated recipient must first derive the common secret K and then use Eq. (14) to obtain the message. We show that the designated recipient can correctly compute the shared secret K with his private key and T, the last element of the received authenticated ciphertext. From the right-hand side of Eq. (12), we have

$$
\begin{aligned}
&(T)^{x_v}\\
&= (g^{(s + \theta)})^{x_v} && \text{(by Eq. (9))}\\
&= y_v^{(s + \theta)}\\
&= K \pmod{p} && \text{(by Eq. (8))}
\end{aligned}
$$

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

If the authenticated ciphertext (Q_1, Q_2, R, T) is correctly generated, it will pass the test of Eq. (15). From the right-hand side of Eq. (15), we have

$$
\begin{aligned}
&g^s y_D^{h_1(m,R)} h_2(m)\\
&= g^{\sum_{U_j \in SO} s_j} g^{d_0 h_1(m,R)} h_2(m) && \text{(by Eq. (7))}\\
&= g^{\sum_{U_j \in SO} s_j + h_1(m,R) \sum_{U_j \in SO} c_j x_j} h_2(m)\\
& && \text{(by Lagrange Interpolation [22])}\\
&= g^{\sum_{U_j \in SO} s_j + e_j h_1(m,R)} h_2(m)\\
&= g^{\sum_{U_j \in SO} r_j} h_2(m) && \text{(by Eq. (5))}\\
&= \prod_{j=1}^{n} R_j h_2(m)\\
&= R \pmod{p} && \text{(by Eq. (4))}
\end{aligned}
$$

which leads to the left-hand side of Eq. (15).

4.3. Variant with Message Linkages

Due to the limited system bandwidth, an online processing system often has difficulty encrypting a large message. For example, for an encryption system such as RSA system which processes message block of 1024 bits, a 1KB message must be divided into 8 message blocks before encryption. In the subsection, we propose a variant with message linkages for facilitating this case by dividing a large message into lots of small message blocks. The construction is similar to that in Section 4.1. We only describe the different parts as follows:

- **Authenticated-Ciphertext-Generation (ACG):**

  For signing the large message m on behalf of the signing group O, each $U_i \in SO$ first divides the message m into z pieces, i.e., $m = m_1 \parallel m_2 \parallel \ldots \parallel m_z$ such that each m_l has a suitable length, and then chooses $r_i \in_R Z_q$ and $w_0 = 0$ to compute c_i, e_i, R_i, R and s_i as Eqs. (1) to (5). The clerk U_{ck} computes s, K, T and Q_1 as Eqs. (7) to (10). U_{ck} further computes

$$w_l = m_l \cdot h_3(w_{l-1} \oplus h_3(K)) \bmod p,$$
$$\text{for } l = 1, 2, \ldots, z, \tag{16}$$

and deliveries $\delta = (Q_1, R, T, w_1, w_2, \ldots, w_z)$ to the designated recipient U_v.

- **Signature-Recovery-and-Verification (SRV):**

  Upon receiving the authenticated ciphertext $\delta = (Q_1, R, T, w_1, w_2, \ldots, w_z)$, U_v first computes K and s as Eqs. (12) and (13). He then computes

  $$m_l = w_l \cdot h_3(w_{l-1} \oplus h_3(K))^{-1} \bmod p,$$
  $$\text{for } l = 1, 2, \ldots, z, \qquad (17)$$

  and recovers the message m as $m_1 \| m_2 \| \ldots \| m_z$. U_v can further verify the multi-signature by checking Eq. (15).

We show that with the authenticated ciphertext $(Q_1, R, T, w_1, w_2, \ldots, w_z)$, the designated recipient U_v can recover the message m and check its validity with Eq. (17). From the right-hand side of Eq. (17), we have

$$w_l \cdot h_3(w_{l-1} \oplus h_3(K))^{-1}$$
$$= m_l \cdot h_3(w_{l-1} \oplus h_3(K)) \cdot h_3(w_{l-1} \oplus h_3(K))^{-1} \quad \text{(by Eq. (16))}$$
$$= m_l \pmod p$$

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

5. Security Proof and Comparison

In this section, we first address the security model with respect to the proposed scheme and prove it in the random oracle model. Then some comparisons with related schemes are made.

5.1. Security Model

Any cryptographic scheme simultaneously satisfying the properties of confidentiality and authenticity should consider the security requirements of 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 ciphertext with respect to two candidate messages. We define these notions as follows:

Definition 3. ***(Confidentiality)*** *A (t, n)-TCAE scheme is said to be semantically secure against adaptive chosen-ciphertext attacks (IND-CCA2) if there exists 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 *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:

- *Authenticated-Ciphertext-Generation (ACG) queries:* $\mathcal{A}$ chooses a message m and then gives $\mathcal{B}$ the message m. $\mathcal{B}$ runs the ACG algorithm on behalf of the signing group and forwards the outputted authenticated ciphertext δ to $\mathcal{A}$.
- *Signature-Recovery-and-Verification (SRV) queries:* $\mathcal{A}$ submits an authenticated ciphertext δ to $\mathcal{B}$. Then $\mathcal{B}$ runs the SRV algorithm on behalf of the designated recipient. If δ is valid, the recovered message m and its converted multi-signature Ω are returned; else, the error symbol ¶ is outputted as a result.

Challenge: The adversary $\mathcal{A}$ produces two messages, m_0 and m_1, of the same length. The challenger $\mathcal{B}$ flips a coin $\lambda \leftarrow \{0, 1\}$ and generates an authenticated ciphertext δ^* for m_λ. The ciphertext δ^* 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 for the SRV query for the target ciphertext.

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 $Adv(\mathcal{A}) = |\Pr[\lambda' = \lambda] - 1/2|$.

Definition 4. ***(Unforgeability)*** *A (t, n)-TCAE scheme is said to achieve existential unforgeability against adaptive chosen message attacks (EF-CMA) if there exists 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 *params* to the adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ adaptively issues ACG queries as those in Phase 1 of Definition 3.

Forgery: Finally, $\mathcal{A}$ arbitrarily chooses a message m and produces an authenticated ciphertext δ^* which is not outputted by the ACG query. The adversary $\mathcal{A}$ wins if δ^* is valid.

5.2. Security Proof

We prove that the proposed scheme achieves the IND-CCA2 and the EF-CMA security in the random oracle model as Theorems 1 and 2, respectively. The security proofs can also be applied to its variant with message linkages, since they have almost the same structure.

Theorem 1. (Proof of Confidentiality) *The proposed scheme is $(t, q_{h_1}, q_{h_2}, q_{h_3}, q_{ACG}, q_{SRV}, \varepsilon)$-secure against adaptive chosen-ciphertext attacks (IND-CCA2) in the random oracle model if there exists no probabilistic polynomial-time adversary that can (t', ε')-break the CDHP, where*

$$\varepsilon' \geq (q_{h_3}{}^{-1})(2\varepsilon - \frac{q_{SRV}(q_{h_1} + q_{h_3} + 1)}{2^k}),$$

$$t' \approx t + t_\lambda(q_{h_2} + 4q_{ACG} + 2q_{SRV}).$$

Here t_λ is the cost for performing a modular exponentiation over a finite field.

Proof: Suppose that a probabilistic polynomial-time adversary $\mathcal{A}$ can (t, q_{h_1}, q_{h_2}, q_{h_3}, q_{ACG}, q_{SRV}, ε)-break the proposed scheme with non-negligible advantage ε under the adaptive chosen-ciphertext attack after running in time at most t and asking at most q_{h_i} h_i random oracle (for i = 1 to 3), q_{ACG} ACG and q_{SRV} SRV queries. Then we can construct another algorithm $\mathcal{B}$ that (t', ε')-breaks the CDHP by taking $\mathcal{A}$ as a subroutine. Let all involved parties and parameters be defined the same as those in Section 4.1. Note that in this proof, the hash functions h_1 to h_3 are simulated as random oracles which must be queried in order to get the output with respect to any input. The objective of $\mathcal{B}$ is to obtain ($g^{d_0 x_v}$ mod p) by taking (p, q, g, y_D, y_v) 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 *params* = {p, q, g, h_1, h_2, h_3} along with (y_D, y_v) to the adversary $\mathcal{A}$.

Phase 1: $\mathcal{A}$ issues the following kinds of queries adaptively:

- h_1 *oracle:* When $\mathcal{A}$ queries an h_1 oracle of $h_1(m, R)$, $\mathcal{B}$ returns **$\mathcal{O}$-Sim_h_1**(m, R). The simulated random oracle **$\mathcal{O}$-Sim_h_1** operates as shown in Fig. 1. Note that the function **insert**(N, b) will insert the value b into the array N.
- h_2 *oracle:* When $\mathcal{A}$ queries an h_2 oracle of $h_2(m)$, $\mathcal{B}$ returns **$\mathcal{O}$-Sim_h_2**(m). The simulated random oracle **$\mathcal{O}$-Sim_h_2** operates as depicted in Fig. 2. Note that the function **check**(N, b) will return a Boolean value depending on whether the value b is stored in the array N or not.

```
oracle O-Sim_h1(m, R)
int Q_h1[$q_{h_1}$][2], A_h1[$q_{h_1}$]; // Let Q_h1 and A_h1 be two arrays.
for i = 0 to $q_{h_1} - 1$
    if (Q_h1[i][0] = m) and (Q_h1[i][1] = R) then
        exit for;  // It is an old query.
    else if (Q_h1[i][0] = “”) then  // It is a new query.
        insert(Q_h1, (m, R)); A_h1[i] ← $v_1 \in_R Z_q$; exit for;
        end if
next i
return A_h1[i];
```

Figure 1. Algorithm of the simulated random oracle **$\mathcal{O}$-Sim_h_1**

```
oracle O-Sim_h2(m)
int Q_h2[$q_{h_2}$], A_h2[$q_{h_2}$][3]; // Let Q_h2 and A_h2 be two arrays.
for i = 0 to $q_{h_2} - 1$
    if (Q_h2[i] = m) then  // It is an old query.
        exit for;
    else if (Q_h2[i] = “”) then  // It is a new query.
        Q_h2[i] ← m; A_h2[i][0] ← $v_2 \in_R Z_q$;
        if (check(Q_h1, m)) = true) then // h1(m, *) has ever been queried.
            for j = 0 to $q_{h_1} - 1$
                if (Q_h1[j][0] = m) then
                    R = Q_h1[j][1]; A_h2[i][1] ← R; A_h2[i][2] ← $V_2 = Rg^{v_2}$ mod p; exit for;
                end if
            next j
        else // h1(m, *) has never been queried.
            A_h2[i][1] = 1; A_h2[i][2] ← $V_2 = g^{v_2}$ mod p;
        end if
        exit for;
    end if
next i
return A_h2[i][2];
```

Figure 2. Algorithm of the simulated random oracle **$\mathcal{O}$-Sim_h_2**

```
oracle $\mathcal{O}$-Sim_$h_3$($K$)
int Q_$h_3$[$q_{h_3}$], A_$h_3$[$q_{h_3}$]; // Let Q_$h_3$ and A_$h_3$ be two arrays.
for $i = 0$ to $q_{h_3} - 1$
    if (Q_$h_3[i] = K$) then  // It is an old query.
        exit for;
    else if (Q_$h_3[i]$ = “”) then  // It is a new query.
        Q_$h_3[i] \leftarrow K$; A_$h_3[i] \leftarrow v_3 \in_R \{0, 1\}^k$; exit for;
    end if
next $i$
return A_$h_3[i]$;
```

Figure 3. Algorithm of the simulated random oracle $\mathcal{O}$-**Sim_*h*3**

```
oracle $\mathcal{O}$-Sim_$ACG$($m$)
$V_2$ = $\mathcal{O}$-Sim_$h_2$($m$);
do
        Choose $s$, $v_1 \in_R Z_q$; $R = g^s {y_D}^{v_1} V_2 \bmod p$ ;
while (check(Q_$h_1$, ($m$, $R$)) = true)
insert(Q_$h_1$, ($m$, $R$)); insert(A_$h_1$, $v_1$);  // define $h_1(m, R) = v_1$
Choose $\theta \in_R Z_q$; $K = {y_v}^{(s + \theta)} \bmod p$; $T = g^{(s + \theta)} \bmod p$; $Q_1 = sh_4(K)$; $Q_2$ = $\mathcal{O}$-Sim_$h_3$($K$) $\oplus m$;
return $\delta = (Q_1, Q_2, R, T)$;
```

Figure 4. Algorithm of the simulated ACG oracle $\mathcal{O}$-**Sim_*ACG***

```
oracle $\mathcal{O}$-Sim_$SRV$($\delta$) // $\delta = (Q_1, Q_2, R, T)$
if (check(Q_$h_1$, $R$) = true) then // $h_1(*, R)$ has ever been queried.
    for $j = 0$ to $q_{h_1} - 1$
        if (Q_$h_1[j][1] = R$) then $m$ = Q_$h_1[j][0]$; exit for;
        end if
    next $j$
    $v_3 = Q_2 \oplus m$;
    if (check(A_$h_3$, $v_3$)) = true) then
        for $j = 0$ to $q_{h_3} - 1$
            if (A_$h_3[j] = v_3$) then $K$ = Q_$h_3[j]$; exit for;
            end if
        next $j$
        $s = Q_1(K \bmod q)^{-1} \bmod q$;
        if ($R = g^s {y_D}^{h_1(m,R)} h_2(m) \bmod p$) then return ($m$, $R$, $s$);
        else
            return ¶;
        end if
    else
        return ¶;
    end if
else // $h_1(*, R)$ has never been queried.
        return ¶;
end if
```

Figure 5. Algorithm of the simulated SRV oracle $\mathcal{O}$-**Sim_*SRV***

```
algorithm Sim_Challenge($m_\lambda$)
    $V_2$ = O-Sim_h2($m_\lambda$);
    do
        Choose $s^*, v_1 \in_R Z_q$; $R^* = g^{s^*} {y_D}^{v_1} V_2 \bmod p$;
    while (check(Q_$h_1$, ($m_\lambda$, $R^*$)) = true)
    insert(Q_$h_1$, ($m_\lambda$, $R^*$)); insert(A_$h_1$, $v_1$); // define $h_1(m_\lambda, R^*) = v_1$
    Choose $\theta, K^{**} \in_R Z_q$, $v_3 \in_R \{0, 1\}^k$;
    $T^* = {y_D}^{(s^* + \theta)} \bmod p$; // implicitly define $K^* = {y_v}^{d_0(s + \theta)} \bmod p$
    // Note that $\mathcal{B}$ does not know $K^*$.
    $Q_1^* = sK^{**}$; // implicitly define $K^{**} = h_4(K^*)$
    $Q_2^* = v_3 \oplus m$; // implicitly define $h_3(K^*) = v_3$
    return $\delta^* = (Q_1^*, Q_2^*, R^*, T^*)$;
```

Figure 6. Algorithm of the simulated **Sim_Challenge**

- *h_3 oracle:* When $\mathcal{A}$ queries an h_3 oracle of $h_3(K)$, $\mathcal{B}$ returns **O-Sim_*h*3**(K). The simulated random oracle **O-Sim_*h*3** operates as shown in Fig. 3
- *ACG queries:* When $\mathcal{A}$ makes an ACG query for some message m, $\mathcal{B}$ returns **O-Sim_*ACG*(*m*)** as the result. The simulated ACG oracle **O-Sim_*ACG*** operates as depicted in Fig. 4.

  SRV queries: When $\mathcal{A}$ makes an SRV query for some authenticated ciphertext δ, $\mathcal{B}$ returns **O-Sim_*SRV*(*δ*)** as the result. The simulated SRV oracle **O-Sim_*SRV*** operates as shown in Fig. 5.

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 $\delta^* = (Q_1^*, Q_2^*, R^*, T^*)$ for m_λ by running the simulated **Sim_Challenge**(m_λ). The algorithm of **Sim_Challenge** operates as depicted in Fig. 6.

Phase 2: $\mathcal{A}$ makes new queries as those stated in Phase 1 except for the SRV query for the target ciphertext δ^*.

Guess: $\mathcal{A}$ outputs a bit λ' as the result.

Output: Finally, $\mathcal{B}$ randomly selects K of the Q_h_3 array and outputs $K^{(s+\theta)^{-1}}$ as a correct answer to the CDHP.

Analysis of the game: Since $\mathcal{B}$ always returns a valid authenticated ciphertext for each issued ACG query without abortion, the simulation of ACG queries is said to be perfect. We then evaluate the simulation of SRV queries. Let SRV_ERR be the event that an SRV query fails during the entire game, i.e., an error symbol is returned for a valid authenticated ciphertext.

An SRV query for some valid $\delta = (Q_1, Q_2, R, T)$ fails if $\mathcal{A}$ can produce δ without asking the corresponding $h_1(m, R)$ or $h_3(K)$ random oracles beforehand. Let AC-V be an event that the authenticated ciphertext δ of an SRV query made by $\mathcal{A}$ is valid. QH$_1$ and QH$_3$ separately denote the events that $\mathcal{A}$ has ever asked $h_1(m, R)$ and $h_3(K)$ random oracles beforehand. Then we can express the fail probability of any SRV query as

$$\begin{aligned}
&\Pr[\text{AC-V} \mid \neg\text{QH}_3] + \Pr[\text{AC-V} \wedge \text{QH}_3 \mid \neg\text{QH}_1] \\
&\quad = \Pr[\text{AC-V} \wedge \text{QH}_1 \mid \neg\text{QH}_3] \\
&\quad + \Pr[\text{AC-V} \wedge \neg\text{QH}_1 \mid \neg\text{QH}_3] \\
&\quad + \Pr[\text{AC-V} \wedge \text{QH}_3 \mid \neg\text{QH}_1] \\
&\quad \le \Pr[\text{QH}_1 \mid \neg\text{QH}_3] \\
&\quad + \Pr[\text{AC-V} \mid \neg\text{QH}_1 \wedge \neg\text{QH}_3] \\
&\quad + \Pr[\text{AC-V} \wedge \text{QH}_3 \mid \neg\text{QH}_1] \\
&\quad \le \frac{q_{h_1}}{2^k} + \frac{1}{2^k} + \frac{q_{h_3}}{2^k} = \frac{q_{h_1} + q_{h_3} + 1}{2^k}.
\end{aligned}$$

Besides, $\mathcal{A}$ can make at most q_{SRV} SRV queries. Consequently, we can express the probability of SRV_ERR as

$$\Pr[\text{SRV\_ERR}] \le \frac{q_{SRV}(q_{h_1} + q_{h_3} + 1)}{2^k}. \tag{18}$$

Also note that in the challenge phase, $\mathcal{B}$ has returned a simulated authenticated ciphertext $\delta^* = (Q_1^*, Q_2^*, R^*, T^*)$ where $T^* = {y_D}^{(s^* + \theta)} \bmod p$, i.e., the value K^* is implicitly defined as $K^* = {y_v}^{d_0(s + \theta)} \bmod p$. As long as the adversary $\mathcal{A}$ never makes an $h_3(K^*)$ query in Phase 2, the entire simulation game could finish without abortion. Let QH$_3$* be the event that $\mathcal{A}$ does make an $h_3(K^*)$ query in Phase 2, and GP the event that the entire simulation game does not abort. When the entire simulation game is normally terminated, it is obvious that $\mathcal{A}$ gains no advantage in guessing λ, i.e.,

$$\Pr[\lambda' = \lambda \mid \text{GP}] = 1/2. \tag{19}$$

We can further rewrite $\Pr[\lambda' = \lambda]$ as

$$\begin{aligned}
\Pr[\lambda' = \lambda] &= \Pr[\lambda' = \lambda \mid \text{GP}] \Pr[\text{GP}] \\
&\quad + \Pr[\lambda' = \lambda \mid \neg\text{GP}] \Pr[\neg\text{GP}]
\end{aligned}$$

$$\le (1/2)\Pr[\mathrm{GP}] + \Pr[\neg\mathrm{GP}] \qquad \text{(by Eq. (19))}$$

$$= (1/2)(1 - \Pr[\neg\mathrm{GP}]) + \Pr[\neg\mathrm{GP}]$$

$$= (1/2) + (1/2)\Pr[\neg\mathrm{GP}]. \qquad (20)$$

Moreover, we can also derive that

$$\Pr[\lambda' = \lambda] \ge \Pr[\lambda' = \lambda \mid \mathrm{GP}]\ \Pr[\mathrm{GP}]$$

$$= (1/2)(1 - \Pr[\neg\mathrm{GP}])$$

$$= (1/2) - (1/2)\Pr[\neg\mathrm{GP}]. \qquad (21)$$

Combing Eqs. (18) and (19), we obtain that

$$|\Pr[\lambda' = \lambda] - 1/2| \le (1/2)\Pr[\neg\mathrm{GP}]. \qquad (22)$$

By the definition of $\mathcal{A}$'s advantage in the security model, we have

$$\varepsilon = |\Pr[\lambda' = \lambda] - 1/2|$$

$$\le (1/2)\Pr[\neg\mathrm{GP}] \qquad \text{(by Eq. (22))}$$

$$= (1/2)(\Pr[\mathrm{QH_3^*} \vee \mathrm{SRV\_ERR}])$$

$$\le (1/2)(\Pr[\mathrm{QH_3^*}] + \Pr[\mathrm{SRV\_ERR}])$$

Rewriting the above inequality, we have

$$\Pr[\mathrm{QH_3^*}] \ge 2\varepsilon - \Pr[\mathrm{SRV\_ERR}])$$

$$\ge 2\varepsilon - \frac{q_{SRV}(q_{h_1} + q_{h_3} + 1)}{2^k}.$$

If the event QH$_3$* happens, we claim that the value $K^* = y_v{}^{d_0(s+\theta)} \bmod p$ will be left in some entry of the Q_h_3 array. Therefore, $\mathcal{B}$ is capable of outputting $(K^*)^{(s+\theta)^{-1}} = g^{d_0 x_v}$ as the correct answer to the CDHP with non-negligible probability

$$\varepsilon' \ge (q_{h_3}{}^{-1})(2\varepsilon - \frac{q_{SRV}(q_{h_1} + q_{h_3} + 1)}{2^k}).$$

The time required for $\mathcal{B}$ is $t' \approx t + t_\lambda(q_{h_2} + 4q_{ACG} + 2q_{SRV})$.

Q.E.D.

To prove that the proposed scheme achieves the EF-CMA security in the random oracle model, we utilize the Forking lemma [24] presented by Pointcheval and Stern. According to their proof techniques, we can directly obtain the same result as follows.

(The Forking Lemma) *Pointcheval and Stern introduced the Forking lemma to prove the security of digital signature schemes. In the random oracle mode, let (G, Σ, V) be a generic signature scheme and A a probabilistic polynomial-time Turing machine whose input only consists of public data. We denote by N1 and N2 the numbers of queries that A can ask to the random oracle and to the signer, respectively. Assume that, within a time bound T, A produces, with probability ε ≥ 10(N2 + 1)(N2 + N1)/2k, a valid signature (m, σ1, h, σ2) where σ1 = R, h = (h1(m, R), h2(m)) and σ2 = s. If the triples (σ1, h, σ2) 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 A replacing interaction with the signer by simulation and produces two valid signatures (m, σ1, h, σ2) and (m, σ1, h', σ2') such that h1(m, R) ≠ h'1(m, R) in the expected time T ' ≤* 120686T/*ε.*

More concretely, in our scheme, we can first obtain two equations below:

$$R = g^s y_D{}^{h_1(m,\,R)} h_2(m) \bmod p,$$

$$R = g^{s'} y_D{}^{h'_1(m,\,R)} h_2(m) \bmod p.$$

Then the private key d_0 can be computed as

$$d_0 = (s - s')/(h'_1(m, R) - h_1(m, R)).$$

Still, to show the tight relation between the security of our proposed scheme and the intractability of the DLP, we give another more detailed security proof as Theorem 2.

Theorem 2. (Proof of Unforgeability) *The proposed scheme is (t, q_{h_1}, q_{h_2}, q_{ACG}, ε)-secure against existential forgery on adaptive chosen-message attacks (EF-CMA) in the random oracle model if there exists no probabilistic polynomial-time adversary that can (t', ε')-break the DLP, where*

$$\varepsilon' \ge (2^{-1})(\varepsilon - 2^{-k})(1 + 4^{-1}(\varepsilon - 2^{-k})^2(2^{-1} + q_{h_1}{}^{-1})),$$

$$t' \approx t + t_\lambda(q_{h_2} + 4q_{ACG}).$$

Here t_λ is the cost for performing a modular exponentiation over a finite field.

Proof: Suppose that a probabilistic polynomial-time adversary $\mathcal{A}$ can $(t, q_{h_1}, q_{h_2}, q_{ACG}, \varepsilon)$-break the proposed scheme with non-negligible advantage ε under the adaptive chosen-message attack after running in time at most t and asking at most q_{h_i} h_i random oracle (for i=1 and 2) and q_{ACG} ACG queries. Then we can construct another algorithm $\mathcal{B}$ that (t', ε')-breaks the DLP by taking $\mathcal{A}$ as a subroutine. Let all involved parties and notations be defined the same as those in Section 4.1, h_3 and h_4 two collision resistant hash functions, and (h_1, h_2) random oracles. The objective of $\mathcal{B}$ is to obtain $d_0 (= \log_g y_D)$ by taking (p, q, g, y_D) 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 to obtain the system's public parameters *params* = $\{p, q, g, h_1, h_2, h_3\}$ and comes up with a random tape composed of a long sequence of random bits. Then $\mathcal{B}$ simulates one or two runs of (t, n)-TCAE scheme to the adversary $\mathcal{A}$ on input *params*, y_D, $y_v = g^\alpha \bmod p$ where $\alpha \in_R Z_q$, and the random tape.

Phase 1: $\mathcal{A}$ adaptively asks h_1 and h_2 random oracles, and ACG queries as those defined in Theorem 1.

Analysis of the game: As the simulated result of each ACG query is computationally indistinguishable from the one produced by a real scheme, the adversary $\mathcal{A}$'s view in the simulation is just like that he is playing in the real scheme. Let AC-V be the event that $\mathcal{A}$ attempts to forge an authenticated ciphertext for his arbitrarily chosen message m and then finally outputs a valid $\delta = (Q_1, Q_2, R, T)$. By assumption, $\mathcal{A}$ has non-negligible probability ε to break the proposed scheme under the adaptive chosen-message attack, i.e., Pr[AC-V] = ε. It is possible that $\mathcal{A}$ successfully generates a valid δ without asking $h_1(m, R)$ and $h_2(m)$ random oracles beforehand. We denote the event that $\mathcal{A}$ guesses correct random values without asking random oracles by NH and we know that Pr[NH] $\leq 2^{-k}$. Then, we can further express the probability that $\mathcal{A}$ outputs a valid forgery $\delta = (Q_1, Q_2, R, T)$ after asking $h_2(m)$ and $h_1(m, R)$ random oracles as

$$\Pr[\text{AC-V} \wedge \neg\text{NH}] \geq (\varepsilon - 2^{-k}).$$

With the outputted $\delta = (Q_1, Q_2, R, T)$, $\mathcal{B}$ first recovers (m, s) using the initially selected private key α, and then checks the entry of A_h_2 array in relation to the $h_2(m)$ query. According to the simulation algorithm of **O-Sim_*h*2**(m), the output of $h_2(m)$ random oracle would be either Rg^{v_2} or g^{v_2}. If $h_2(m) = Rg^{v_2}$, we have

$$R = g^s y_D^{h_1(m,\, R)} h_2(m) \bmod p$$

$$= g^s y_D^{h_1(m,\, R)} R g^{v_2} \bmod p.$$

Rewriting the above equality, we can obtain $g^{-v_2} = g^s y_D^{h_1(m,\, R)} \bmod p$. Then $\mathcal{B}$ will be able to solve the DLP by computing

$$d_0 = (-v_2 - s)h_1(m, R)^{-1} \bmod q.$$

Since the supplied sequence of random bits are unpredictable and each random oracle is simulated without collision, we can expect that $\Pr[h_2(m) = Rg^{v_2}] = 2^{-1}$, i.e., $\mathcal{B}$ would have the probability of 2^{-1} to solve the DLP in the first simulation on condition that the event (AC-V $\wedge$ ¬NH) happens.

In the other hand, with the probability of $(1 - 2^{-1}) = 2^{-1}$, $\mathcal{B}$ might have to launch the second simulation in case that $h_2(m) = g^{v_2}$. $\mathcal{B}$ again runs $\mathcal{A}$ on input *params*, y_D, $y_v = g^{\alpha} \bmod p$ where $\alpha \in_R Z_q$, and the same random tape. As the adversary $\mathcal{A}$ is given the same sequence of random bits, we know that the i-th random query he makes will always be the same as the one during the first simulation. For all the oracle queries before the $h_1(m, R)$ query, $\mathcal{B}$ returns identical results as those in the first time. When $\mathcal{A}$ asks an $h_1(m, R)$ random oracle, $\mathcal{B}$ directly gives a new $v_1^* \in_R Z_q$ instead of v_1. At the same time, $\mathcal{A}$ is provided with another different random tape which is also composed of a long sequence of random bits. By the "Forking lemma" stated above, when $\mathcal{A}$ eventually outputs another valid authenticated ciphertext $\delta^* = (Q_1^*, Q_2^*, R, T^*)$ with $h_1(m, R) \neq h_1^*(m, R)$ or $h_2(m) = Rg^{v_2}$ this time, $\mathcal{B}$ would have a chance to solve the DLP. To evaluate $\mathcal{B}$'s success probability, we use the "Splitting lemma" [24] as follows:

Let X and Y be the sets of possible sequences of random bits and random function values supplied to $\mathcal{A}$ before and after the $h_1(m, R)$ query is made, respectively. It follows that on inputting a random value $(x \parallel y)$ for any $x \in X$ and $y \in Y$, $\mathcal{A}$ outputs a valid forgery with the probability of ε, i.e., $\Pr_{x \in X,\, y \in Y}$[AC-V] = ε. 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). $\forall x \in D$, $\Pr_{y \in Y}[\text{AC-V}] \geq 2^{-1}\varepsilon$.

From the above definition, we can derive that if $\rho \in D$ is the supplied sequence of random bits and random function values for $\mathcal{A}$ before the $h_1(m, R)$ query is made, then for any sequence of random bits and random function values $y' \in Y$ after that, $\mathcal{A}$ outputs a valid forgery in the second simulation with the probability of at least $(2^{-1}\varepsilon)^2 = 4^{-1}\varepsilon^2$, i.e., $\Pr_{\rho \in D,\, y' \in Y}[\text{AC-V}] \geq 4^{-1}\varepsilon^2$. Since $\Pr[h_2(m) = Rg^{v_2}] = 2^{-1}$ and the probability that $\mathcal{A}$ outputs another $\delta^* = (Q_1^*, Q_2^*, R, T^*)$ with $h_1(m, R) \neq h_1^*(m, R)$ is ${q_{h_1}}^{-1}$, we can express the probability that $\mathcal{B}$ solves the DLP in the second simulation as

$$(\varepsilon - 2^{-k})(4^{-1}(\varepsilon - 2^{-k})^2)(2^{-1} + {q_{h_1}}^{-1})$$

$$= 4^{-1}(\varepsilon - 2^{-k})^3(2^{-1} + {q_{h_1}}^{-1}).$$

Combining the result in the first simulation, we can derive that after the second simulation, $\mathcal{B}$ could solve the DLP with the success probability

$$\varepsilon' \geq (2^{-1})(\varepsilon - 2^{-k})$$

$$+ (1 - 2^{-1})(4^{-1}(\varepsilon - 2^{-k})^3(2^{-1} + {q_{h_1}}^{-1}))$$

$$= (2^{-1})(\varepsilon - 2^{-k})(1 + 4^{-1}(\varepsilon - 2^{-k})^2(2^{-1} + {q_{h_1}}^{-1})).$$

In addition, the time required for $\mathcal{B}$ in one simulation is

$$t + t_\lambda(q_{h_2} + 4q_{ACG}).$$

We hence can represent the total time for $\mathcal{B}$ after the second simulation as

$$t' \approx (2^{-1})(t + t_\lambda(q_{h_2} + 4q_{ACG}))$$

$$+ (1 - 2^{-1})(t + t_\lambda(q_{h_2} + 4q_{ACG}))$$

$$= t + t_\lambda(q_{h_2} + 4q_{ACG}).$$

Q.E.D.

According to Theorem 2, the proposed scheme is secure against existential forgery attack, which stands for that the signing group cannot deny having generated their authenticated ciphertext. Hence, we obtain the following corollary.

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

5.3. Comparisons

To the best of our knowledge, the proposed scheme is the first provably secure (t, n)-TCAE scheme based on the CDHP. We compare the proposed scheme with some related works including Lv *et al.*'s (LWK for short) [15], Tso *et al.*' (TOO for short) [25], Araki *et al.*' (AUI for short) [9], the Wu-Hsu (WH for short) [10], Wu *et al.*'s (WHT for short) [18], Chang's (Cha for short) [20], Tsai's (Tsa for short) [19] and the Lin-Yeh (LY for short) [21] schemes in terms of functionalities and security proofs. Detailed comparisons are demonstrated as Table 1.

To evaluate the computational performance, we further compare the proposed scheme with above group-oriented CAE ones in number of the most time-consuming operation, i.e., modular exponentiation computation. The required computational costs with respect to each compared scheme are shown as Table 2. From this table, it can be seen that if we let $t \approx n$, Tsai's scheme would have the same performance as ours. Yet, in practice, our scheme with threshold signing group would offer much more flexibility than Tsai's. To sum up, we claim that the proposed scheme provides not only better functionalities, but also lower computational costs.

6. Conclusion

In this paper, we have proposed a secure (t, n)-TCAE scheme for confidential applications in the multi-user environment. A variant with message linkages is also introduced for facilitating the situation where a large message needs to be transmitted over the public network. Both the proposed scheme and its variant can simultaneously satisfy the security requirements of authenticity, confidentiality and non-repudiation. Unlike previous schemes focusing on the single-user setting, our proposed scheme and its variant allow any t or more signers to cooperatively generate a valid authenticated ciphertext. The conversion mechanism enables the designated recipient to publicize the signing group's ordinary multi-signature for protecting his benefits. Furthermore, we also proved that the proposed scheme achieves the IND-CCA2 and the EF-CMA security in the random oracle model. Compared with related works, ours provides not only better functionalities, but also lower computational costs.

Table 1. Comparisons in terms of functionalities and security proofs

Scheme / Item	LWK	TOO	AUI	WH	WHT	Cha	Tsa	LY	Ours
Multi-User Environment	×	×	×	×	O	O	O	O	O
Threshold Groups	×	×	×	×	×	O	×	O	O
Message Linkages	O	×	×	×	×	×	×	×	O
Signature Conversion	O	O	O	O	O	O	O	O	O
No Conversion Cost	O	O	×	O	O	O	O	O	O
Proof of Confidentiality	×	×	×	×	O	O	×	×	O
Proof of Unforgeability	×	×	×	×	O	O	×	×	O

Table 2. Comparisons in number of required modular exponentiation operations

Scheme / Item	WHT	Cha	Tsa	LY	Ours
Computational Costs*	$3n^2 - n + 5$	$3n^2 - n + 5$	$3n + 5$	$3t^2 + t + 3$	$3t + 5$

Remark *: Let t be the threshold value and n the size of signing group. The computational costs include those executed by each signer in the signing group, the clerk and the designated recipient.

Acknowledgments

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 103-2221-E-019-001.

References

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

[2] **H. Delfs and H. Knebl.** Introduction to Cryptography: Principles and Applications. *Springer-Verlag, Berlin*, 2002.

[3] **C. L. Hsu, H. Y. Lin.** An identity-based key-insulated encryption with message linkages for peer-to-peer communication network. *KSII Transactions on Internet and Information Systems*, 2013, Vol. 7, No. 11, 2928-2940.

[4] **H. Y. Lin.** Secure universal designated verifier signature and its variant for privacy protection. *Information Technology and Control*, 2013, Vol. 42, No. 3, 268-276.

[5] **H. Y. Lin, T. S. Wu, S. K. Huang.** An efficient strong designated verifier proxy signature scheme for electronic commerce. *Journal of Information Science and Engineering*, 2012, Vol. 28, No. 4, 771-785.

[6] **F. Hou, Z. Wang, Y. Tang, Z. Liu.** Protecting integrity and confidentiality for data communication. In: *Proceedings of the 9th International Symposium on Computers and Communications (ISCC)*, Alexandria, Egypt, 2004, Vol. 1, No. 28, pp. 357-362.

[7] **W. Stallings.** Cryptography and Network Security: Principles and Practices. *4th. Ed., Pearson*, 2005.

[8] **P. Horster, M. Michels, H. Petersen.** Authenticated encryption schemes with low communication costs. *Electronics Letters*, 1994, Vol. 30, No. 15, 1212-1213.

[9] **S. Araki, S. Uehara, K. Imamura.** The limited verifier signature and its application. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 1999, Vol. E82-A, No. 1, 63-68.

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

[11] **B. Meng, S. Wang, Q. Xiong.** A fair non-repudiation protocol. In: *Proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design*, Brazil, 2002, pp. 68-73.

[12] **Y. H. Chen, J. K. Jan.** Enhancement of digital signature with message recovery using self-certified public keys and its variants. *ACM SIGOPS Operating Systems Review*, 2005, Vol. 39, No. 3, 90-96.

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

[14] **Y. Q. Peng, S. Y. Xie, Y. F. Chen, R. Deng, L. X. Peng.** A publicly verifiable authenticated encryption scheme with message linkages. In: *Proceedings of the 3rd International Conference on Networking and Mobile Computing (ICCNMC2005)*, China, 2005, pp. 1271-1276.

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

[16] **M. S. Hwang, C. Y. Liu.** Authenticated encryption schemes: current status and key issues. *International Journal of Network Security*, 2005, Vol. 1, No. 2, 61-73.

[17] **M. S. Hwang, T. Y. Chang.** Threshold signatures: current status and key issues. *International Journal of Network Security*, 2005, Vol. 1, No. 3, 123-137.

[18] **T. S. Wu, C. L. Hsu, K. Y. Tsai, H. Y. Lin, T. C. Wu.** Convertible multi-authenticated encryption scheme. *Information Sciences*, 2008, Vol. 178, No. 1, 256-263.

[19] **J. L. Tsai.** Convertible multi-authenticated encryption scheme with one-way hash function. *Computer Communications*, 2009, Vol. 32, No. 5, 783-786.

[20] **T. Y. Chang.** A convertible multi-authenticated encryption scheme for group communications. *Information Sciences*, 2008, Vol. 178, No. 17, 3426-3434.

[21] **H. Y. Lin, Y. S. Yeh.** A novel (t, n) threshold convertible authenticated encryption scheme. *Applied Mathematical Sciences*, 2008, Vol. 2, No. 5, 249-254.

[22] **C. L. Hsu, H. Y. Lin.** New identity-based key-insulated convertible multi-authenticated encryption scheme. *Journal of Network and Computer Applications*, 2011, Vol. 34, No. 5, 1724-1731.

[23] **B. Wendroff.** Theoretical Numerical Analysis. Academic Press Inc., 1996.

[24] **D. Pointcheval, J. Stern.** Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 2000, Vol. 13, 361-396.

[25] **R. Tso, T. Okamoto, E. Okamoto.** An improved signcryption scheme and its variation. In: *Proceedings of the 4th International Conference on Information Technology (ITNG '07)*, Las Vegas, U.S.A., 2007, pp. 772-778.

Received November 2013.