Efficient Certificate-Based Encryption Scheme Secure against Key Replacement Attacks in the Standard Model

YANG LU AND JIGUO LI
College of Computer and Information Engineering
Hohai University
Nanjing, 211100 P.R. China

Certificate-based encryption is a useful primitive that combines traditional public key encryption and identity-based encryption while preserving some of their most attractive features. It not only simplifies the cumbersome certificate management in traditional PKI but also solves the key escrow problem inherent in identity-based encryption. In this paper, we propose a new certificate-based encryption scheme without random oracles that is provably secure against key replacement attacks. The proposed certificate-based encryption scheme is proven to be secure under the hardness of the decisional 3-Party Diffie-Hellman problem in the standard model. Performance comparison shows that the proposed scheme outperforms all the previous standard-model certificate-based encryption schemes in the literature.

Keywords: public key encryption, certificate-based encryption, key replacement attack, standard model

1. INTRODUCTION

Public key cryptography (PKC) is an important technique to realize network and information security. In traditional PKC, cryptographic keys are generated randomly with no connection to users’ identities. The association between a user’s identity and his public key is obtained through a digital certificate issued by a trusted certificate authority (CA). This kind of certificate systems is referred as public key infrastructure (PKI). However, the need for PKI-certificates is considered as the main difficulty in the deployment and management of traditional PKC. To simplify the certificate management, Shamir [1] introduced the concept of identity-based cryptography (IBC). In IBC, a user’s public key is derived directly from his identity and his private key is generated by a trusted third party called Private Key Generator (PKG). The main practical benefit of IBC lies in the reduction of the need for certificates. However, if the KGC becomes dishonest, it can impersonate any user using its knowledge of the user’s private key. This is due to the key escrow problem inherent in IBC. In addition, private key distribution in identity-based cryptography is a very daunting task as private keys must be sent to the users over secure channels.

In Asiacrypt 2003, Al-Riyami and Paterson [2] proposed the notion of certificateless public key cryptography (CL-PKC). In CL-PKC, a trusted third party called Key Generation Center (KGC) is involved in the process of issuing a partial private key for each user. Every user in the system independently generates a pair of secret key and public key, and then combines his secret key with the partial private key

* This work is supported by the National Natural Science Foundation of China (61272542), the Six Talent Peaks Program of Jiangsu Province of China (2009182) and the Program for New Century Excellent Talents in Hohai University.
from the KGC to generate his private key. This way, the KGC does not know the private key of any user. Therefore, CL-PKC overcomes the key escrow problem inherent in IBC. However, due to the lack of certificates to ensure the authenticity of the public keys, cryptographic schemes in the certificateless setting are susceptible to the key replacement attack. Furthermore, as partial private keys should be sent to the users over secure channels, CL-PKC suffers from the key distribution problem.

In Eurocrypt 2003, Gentry [3] introduced the notion of certificate-based encryption (CBE) that represents an interesting and potentially useful balance between traditional PKC and IBC. As in traditional PKC, each user in a CBE system generates his own public/private key pair and requests a certificate from a CA. The difference is that a certificate in CBE acts not only as a certificate (as in traditional PKC) but also as a partial private key (as in CL-PKC). This additional functionality provides an efficient implicit certificate mechanism so that a receiver needs both his private key and certificate to decrypt a ciphertext sent to him, while message senders need not be concerned about the receiver’s certificate revocation problem. The feature of implicit certificate allows us to eliminate third-party queries for the certificate status and simplify the public key revocation in traditional PKI. As a result, CBE does not need infrastructures like certificate revocation list (CRL) or online certificate status protocol (OCSP). Furthermore, CBE overcomes the key escrow problem (since the CA does not know the private keys of users) and the key distribution problem (since the certificates need not be kept secret) inherent in IBC. In parallel to CBE, Kang et al. [4] proposed the notion of certificate-based signature (CBS) that follows the idea of CBE presented by Gentry. In the recent years, the topic of certificate-based cryptography (CBC) has attracted much attention in the research community and a number of schemes have been proposed, including many CBE schemes (e.g. [5-11]) and CBS schemes (e.g. [12-15]).

The following table summarizes the comparison of the above cryptosystems.

<table>
<thead>
<tr>
<th>Cryptosystem</th>
<th>Do not require trusted third party</th>
<th>Implicit Certificates</th>
<th>Key Escrow Free</th>
<th>Key Distribution Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional PKC</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>IBC</td>
<td>×</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>CL-PKC</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>CBC</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Key replacement attack, which was first introduced to the security of certificateless public key encryption (CL-PKE) by Al-Riyami and Paterson [2], refers to that an adversary replaces a user’s public key with a false one of its choice and dupes any other third party to encrypt messages or verify signatures using this false public key. At the first glance, one may think that this kind of attack does not exist in the certificate-based cryptosystems due to the use of certificates. However, it does exist. In a certificate-based cryptosystem, the CA does issue the certificates. But, only the holder of a certificate needs to check the validity of its certificate and other users do not need. Therefore, the certificate-based cryptosystems are also susceptible to the key replacement attack. Actually, some previous certificate-based schemes (including CBE, CBS and certificate-based signcryption) have been demonstrated to be insecure under this attack. Please refer to [13, 15-17] for the concrete key replacement attacks on those schemes.
Key replacement attacks in CBE systems were first introduced in [16] and formally defined by Lu and Li in [11]. The first CBE scheme that has been explicitly proven secure against key replacement attacks was also proposed in [11]. However, the provable security goal of this scheme was obtained by considering the random oracle model [18]. It is well known that provable security is one of the basic requirements for PKC. As shown in [19], a proof in the random oracle model can only serve as a heuristic argument and does not necessarily imply the security in the real implementation. Therefore, to develop a CBE scheme that is provably secure against key replacement attacks without using the random oracle methodology becomes an important and interesting research problem.

In this paper, we newly propose a CBE scheme based on Waters’ IBE scheme [20]. We prove its security under the hardness of the decision 3-Party Diffie-Hellman problem in the standard model. Compared with the previous standard-model CBE schemes in the literature, our scheme enjoys better computation efficiency and lower communication bandwidth. To the best of our knowledge, our scheme is the first CBE scheme without random oracles that has been explicitly proven secure against key replacement attacks.

The rest of the paper is organized as follows. In Section 2, some related preliminaries are briefly reviewed. The proposed CBE scheme is described in Section 3 and analyzed in Section 4. Finally, a conclusion is given in Section 5.

2. PRELIMINARIES

In this section, we briefly review some preliminaries that are related to our paper.

2.1 Bilinear Map and Computational Assumption

Let $k$ be a security parameter and $p$ be a $k$-bit prime number, $G$ and $G_T$ denote two multiplicative cyclic groups of the same order $p$. A mapping $e: G \times G \rightarrow G_T$ is called a bilinear map if it satisfies the following properties:

- Bilinearity: $e(u^a, v^b) = e(u, v)^{ab}$ for all $u, v \in G$ and $a, b \in \mathbb{Z}_p^*$.
- Non-degeneracy: $e(g, g) \neq 1$ for a random generator $g \in G$.
- Computability: $e(u, v)$ can be efficiently computed for all $u, v \in G$.

The security of our CBE scheme is based on the following Decision 3-Party Diffie-Hellman (3-DDH) problem which was proposed for the first time in [21] as a non-paring variant of the decisional bilinear Diffie-Hellman (DBDH) problem.

**Definition 1.** The 3-DDH problem in $G$ is, given $(g, g^a, g^b, g^c, T) \in G^5$ for unknown $a, b, c \in \mathbb{Z}_p^*$, to decide whether $T = g^{abc}$. The advantage of a probabilistic polynomial time (PPT) algorithm $A$ in solving the 3-DDH problem is defined as $Adv_{A,3-DDH}^{3-DDH}(k) = |\Pr[A(g, g^a, g^b, g^c, T) = 1] - \Pr[A(g, g^a, g^b, g^c, T) = 1]|$, where the probability is defined over the randomly chosen $a, b, c$ and $T$. We say that the 3-DDH problem is hard in $G$ if $Adv_{A,3-DDH}^{3-DDH}(k)$ is negligible for all PPT algorithms $A$. 
2.2 Certificate-Based Encryption

Formally, a CBE scheme consists of the following five algorithms:

- **Setup**($1^k$): On input a security parameter $k$, this algorithm generates a master key $msk$ and a list of public parameters $params$. After this algorithm is performed, the CA publishes $params$ and keeps $msk$ secret.

- **UserKeyGen**(params): On input params, this algorithm outputs a public/private key pair ($PK, SK$). Specifically, when a user with identity $id$ runs this algorithm, the generated key pair is denoted as $(PK_{id}, SK_{id})$.

- **CertGen**(params, $msk$, $id$, $PK_{id}$): On input params, $msk$, and an user’s identity $id$ and public key $PK_{id}$, this algorithm outputs a certificate $Cert_{id}$. After this algorithm is performed, the CA sends the certificate $Cert_{id}$ to the user $id$ via an open channel.

- **Encrypt**(params, $M$, $id$, $PK_{id}$): On input params, a message $M$, and a receiver’s identity $id$ and public key $PK_{id}$, this algorithm outputs a ciphertext $C$.

- **Decrypt**(params, $C$, $SK_{id}$, $Cert_{id}$): On input params, a ciphertext $C$, and the receiver’s private key $SK_{id}$ and certificate $Cert_{id}$, this algorithm outputs either a plaintext $M$ or a special symbol $\bot$ indicating a decryption failure.

The essential security of a CBE scheme requires that one can decrypt a valid ciphertext generated using the public key $PK_{id}$ if and only if he has the knowledge of both $SK_{id}$ and $Cert_{id}$. In other words, one cannot recover the correct plaintext from a valid ciphertext with only $SK_{id}$ or $Cert_{id}$. As defined in [11], the security model of CBE schemes that are secure against key replacement attacks should distinguish two different types of adversaries: Type-I and Type-II. Type-I adversary simulates an uncertified user who wants to gain some information about a message from its encryption before obtaining a certificate from the CA. Such an adversary can replace public keys of any users, but is not allowed to access the master key. Type-II adversary simulates an honest-but-curious CA in possession of the master key who attacks a given public key without the knowledge of the corresponding private key. Such an adversary is able to produce certificates for any users, but is not allowed to replace any user’s public key. Note that if a Type-II adversary is able to replace a user’s public key, then it may trivially break the confidentiality of the communications for that user by performing a man-in-the-middle attack.

The chosen-ciphertext security for CBE schemes is defined by the following two different games “IND-CBE-CCA-I” and “IND-CBE-CCA-II”, in which a Type-I adversary and a Type-II adversary interact with a challenger respectively.

**IND-CBE-CCA-I**: This game is played between a Type-I adversary $\mathcal{A}$ and a challenger. **Setup.** The challenger runs the algorithm **Setup**($1^k$) to generate $msk$ and $params$. It then returns $params$ to $\mathcal{A}$ and keeps $msk$ to itself. **Phase 1.** In this phase, $\mathcal{A}$ can adaptively query the following five oracles:

- **O’CreateUser.** On input an identity $id$, the challenger returns a public key $PK_{id}$. If $id$ has no associated public key, then the challenger runs **UserKeyGen**(params) to generate a key-pair ($SK_{id}, PK_{id}$) and returns $PK_{id}$. In this case, $id$ is said to be created. Note that other oracles defined below only respond to an identity which has been created.
CERTIFICATE-BASED ENCRYPTION SCHEME SECURE AGAINST KEY REPLACEMENT ATTACKS

- $O_{\text{ReplacePublicKey}}$: On input an identity $id$ and a value $PK_{id}'$, the challenger replaces the current public key $PK_{id}$ associated with $id$ with the value $PK_{id}'$. Note that the current value of an entity's public key is used by the challenger in any computations or responses to the following queries. This oracle models the ability of a Type-I adversary to convince a legitimate user to use a false public key.
- $O_{\text{ExtractPrivateKey}}$: On input an identity $id$, the challenger responds with a private key $SK_{id}$. Here, $A$ is disallowed to query this oracle on any identity for which the public key has been replaced. This restriction is imposed due to the fact that it is unreasonable to expect the challenger to be able to provide a private key of a user for which it does not know the private key.
- $O_{\text{GenerateCertificate}}$: On input an identity $id$, the challenger responds with a certificate $Cert_{id}$. If $id$ has no associated certificate, then the challenger should first generate a certificate by running the algorithm $\text{CertGen}(\text{params}, msk, id, PK_{id})$.
- $O_{\text{StrongDecrypt}}$: On input an identity $id$ and a ciphertext $C$, the challenger responds with the decryption of the ciphertext $C$. Note that the challenger should return the correct decryption of $C$ even if the public key associated with $id$ has been replaced. This is a rather strong property for the security model of CBE. After all, the challenger may no longer know the correct corresponding private key. However, this capability may give $A$ more power in breaking a CBE scheme. In addition, if we do not give $A$ access to such a strong decryption oracle, then the ability to replace public keys is of no use to such an adversary. This is because if the challenger never gives a response based on a replaced public key, then $A$ gains no advantage by replacing a public key.

**Challenge.** Once $A$ decides that Phase 1 is over, it outputs an identity $id^*$ and two equal-length messages $(M_0, M_1)$. The challenger first picks a random bit $b \in \{0, 1\}$ and computes the challenge ciphertext $C^* = Encrypt(\text{params}, id^*, PK_{id^*})$. It then outputs $C^*$ to $A$.

**Phase 2.** In this phase, $A$ can issue a second sequence of queries as in Phase 1.

**Guess.** After all queries, $A$ outputs a guess $b' \in \{0, 1\}$ for $b$. We say that $A$ wins the game if $b = b'$ and the following restrictions are simultaneously satisfied: (1) $A$ cannot submit $id'$ to $O_{\text{GenerateCertificate}}$ at any point; (2) $A$ cannot submit an identity to $O_{\text{ExtractPrivateKey}}$ if its corresponding public key has been replaced; (3) $A$ cannot submit $(id^*, C^*)$ to $O_{\text{StrongDecrypt}}$ in Phase 2. $A$'s advantage in the above game is defined to be $Adv_A^{\text{IND-CBE-CCA-II}}(k) = \Pr[b = b'] - 1/2$.

**IND-CBE-CCA-II:** This game is played between a Type-II adversary $A$ and a challenger.

**Setup.** The challenger runs the algorithm $\text{Setup}(1^k)$ to generate $(msk, \text{params})$ and the algorithm $\text{UserKeyGen}(\text{params})$ to obtain a key pair $(PK', SK')$ respectively. It then returns $\text{params}$, $msk$ and $PK'$ to $A$.

**Phase 1.** In this phase, $A$ can adaptively query the following decryption oracle:
- $O_{\text{Decrypt}}$: On input an identity $id$ and a ciphertext $C$, the challenger performs the algorithm $\text{Decrypt}(\text{params}, C, SK')$ and then returns the decrypted plaintext (after running the algorithm $\text{CertGen}(\text{params}, msk, id, PK')$ if necessary).

**Challenge.** Once $A$ decides that Phase 1 is over, it outputs an identity $id^*$ and two equal-length messages $(M_0, M_1)$. The challenger picks a random bit $b \in \{0, 1\}$ and
computes the challenge ciphertext $C' = \text{Encrypt}(\text{params}, M_b, id', PK^*)$. It then outputs $C'$ to $A$.

**Phase 2.** In this phase, $A$ can issue a second sequence of queries as in Phase 1.

**Guess.** After all queries, $A$ outputs a guess $b' \in \{0, 1\}$ for $b$. We say that $A$ wins the game if $b = b'$ and $A$ has never submitted $(id', C')$ to $O^{\text{Encrypt}}$ in Phase 2. $A$’s advantage in the above game is defined to be

$$\text{Adv}_{A}^{\text{IND-CBE-CCA-II}}(k) = |\Pr[b = b'] - 1/2|.$$ 

**Definition 2.** A CBE scheme is said to be IND-CBE-CCA secure if no PPT adversary has non-negligible advantage in the above two games.

### 2.3 Data Encapsulation Mechanism

A data encapsulation mechanism (DEM) is a symmetric encryption scheme that is specified by two algorithms $\text{Encrypt}$ and $\text{Decrypt}$. Encryption algorithm $\text{Encrypt}$ takes as input a message $M \in \{0, 1\}^*$ and a symmetric key $K \in \mathcal{K}$, and outputs a ciphertext $C \in \{0, 1\}^*$, where $\mathcal{K}$ is the symmetric key space. Decryption algorithm $\text{Decrypt}$ takes as input a ciphertext $C$ and a symmetric key $K$ and returns either a decrypted plaintext $M \in \{0, 1\}^*$ or a special symbol $\bot$ denoting a decryption failure.

For the purposes of this paper, it is required that a DEM is secure with respect to indistinguishability against adaptive chosen-ciphertext attacks (IND-DEM-CCA).

**Definition 3.** A DEM is said to be IND-DEM-CCA secure if no PPT adversary $A$ has non-negligible advantage in the following game:

- **Initial.** $A$ runs on input a security parameter $1^k$ and submits two equal-length messages $M_0$ and $M_1$.
- **Challenge.** The challenger randomly chooses a key $K \in \mathcal{K}$ as well as a random bit $b \in \{0, 1\}$, and returns $C' = \text{Encrypt}(M_b, K)$ as a challenge ciphertext to $A$.
- **Query.** In this phase, $A$ can adaptively make a series of queries to a decryption oracle and the challenger responds these queries by using the key $K$. The only restriction is that $A$ cannot submit the challenge ciphertext $C'$ to this decryption oracle.
- **Guess.** $A$ outputs a guess $b' \in \{0, 1\}$ for $b$ and wins the game if $b = b'$. $A$’s advantage in the above game is defined to be

$$\text{Adv}_{A}^{\text{IND-DEM-CCA}}(k) = |\Pr[b = b'] - 1/2|.$$ 

An IND-DEM-CCA secure DEM can be built from relatively weak primitives, i.e. from any passive secure symmetric encryption scheme by essentially adding a message authentication code (MAC). In addition, the modes of operation CMC [22] and EME [23] both give IND-DEM-CCA secure DEMs provided that the underlying block-cipher is a strong pseudorandom permutation and avoid the usual overhead due to the MAC.

### 2.4 Collision Resistant Hash Function

In our CBE construction, we require two collision resistant hash functions.

**Definition 4.** A hash function $H \leftarrow_{\mathcal{R}} \mathcal{H}(k)$ is collision resistant if for all PPT algorithms $A$
3. OUR PROPOSED CBE SCHEME

In this section, we propose a new CBE scheme without random oracles. Our scheme is constructed from Waters’ IBE scheme [20]. In order to perform secrecy communication for arbitrarily long messages, our construction adopts the KEM-DEM hybrid encryption framework introduced by Cramer and Shoup [24]. Our proposed scheme consists of the following five algorithms:

- **Setup**: Let \( G \) and \( G_f \) be two cyclic groups of prime order \( p \) and \( g \) be a generator of \( G \). Given a bilinear map \( e : G \times G \rightarrow G_f \), two collision resistant hash functions \( H_1 : \{0, 1\}^* \times G \rightarrow \{0, 1\}^* \) and \( H_2 : G \rightarrow Z_p^* \), and an IND-DEM-CCA secure data encapsulation mechanism \( \text{DEM} = (\text{Encrypt, Decrypt}) \) with key space \( G_f \). The CA randomly chooses \( \alpha \in Z_p^* \) and computes \( g_1 = g^\alpha \). Additionally, the CA chooses three random elements \( g_2 \), \( g_3 \), \( u \in G \) and a vector \( U = (u_1, u_2, \ldots, u_n) \) whose elements are chosen from \( G \) at random. The public parameters are \( \text{params} = (p, G, G_f, e, g_1, g_2, g_3, u, U, H_1, H_2, \text{DEM}) \) and the master key is \( \text{msk} = \alpha \). For simplicity, given a \( n \)-bit string \( \lambda = \lambda_1 \lambda_2 \cdots \lambda_n \), we define Waters hash function \( F : \{0, 1\}^n \rightarrow G^* \) as \( F(\lambda) = u \prod_{i=1}^n u_i^{\lambda_i} \).

- **UserKeyGen**: A user with identity \( id \) chooses a random value \( x \in Z_p^* \) as his private key \( SK_{id} \) and computes the corresponding public key as \( PK_{id} = (PK_{id}^{(1)}, PK_{id}^{(2)}) = (g^x, g_1^x) \).

- **CertGen**: To generate a certificate for a user with identity \( id \) and public key \( PK_{id} \), the CA chooses a random value \( s \in Z_p^* \) and computes \( Cert_{id} = (Cert_{id}^{(1)}, Cert_{id}^{(2)}, Cert_{id}^{(3)}) = (g_2^s \cdot F(ID)^y, g^x, g_1^x) \), where \( ID = H_1(id, PK_{id}) \).

- **Encrypt**: To send a message \( M \in \{0, 1\}^* \) to a receiver with identity \( id \) and public key \( PK_{id} \), the sender first chooses a random value \( r \in Z_p^* \) and computes a symmetric key \( K = e(PK_{id}^{(1)}, g_2^r) \). It then computes the ciphertext as \( C = (C_1, C_2, C_3) = (g^s, (F(ID) \cdot g_1^x)^y, \text{DEM}.\text{Encrypt}(M, K)) \), where \( ID = H_1(id, PK_{id}) \) and \( t = H_2(C_1) \).

- **Decrypt**: To decrypt a ciphertext \( C = (C_1, C_2, C_3) \), the receiver \( id \) first recovers the symmetric key as \( K' = e(C_{id}, (Cert_{id}^{(1)})^{g_1^x}) \cdot e((Cert_{id}^{(1)\prime})^{SK_{id}} \cdot C_2) \), where \( t = H_2(C_1) \). It then computes \( M/\perp = \text{DEM}.\text{Decrypt}(C_3, K) \).

Note that the above algorithm **Encrypt** does not require any pairing computations once \( e(PK_{id}^{(1)}, g_2) \) has been pre-computed. In addition, the consistency of our scheme is easy to check as we have

\[
K = e(C_{id}, (Cert_{id}^{(1)})^{(Cert_{id}^{(1)\prime})^{SK_{id}}}) \cdot e((Cert_{id}^{(2)})^{SK_{id}}, C_2)
= e(g^x, (g_2^s \cdot F(ID)^y \cdot g_1^x)^y) \cdot e((g^x)^y, (F(ID) \cdot g_1^x)^y)
\]
\begin{align*}
= e(g', g_2^{x'}) & \cdot e(g', F(ID)^{y'}) \cdot e(g', g_3^{y'}) \cdot e(g^{x''}, F(ID)^y) \cdot e(g^{x''}, g_3^y) \\
= e(g^u, g_2^v) & = e(PK_i^{(2)}, g_2)^y.
\end{align*}

4. ANALYSIS OF THE SCHEME

4.1 Security

The security analysis of our scheme requires the following lemma from [25].

**Lemma 1.** Let \( C \) be some “error event” such that \( A|\sim C \) occurs if and only if \( B|\sim C \) occurs. Then \( |\Pr[A] - \Pr[B]| \leq \Pr[C] \).

**Theorem 1.** Our CBE scheme is IND-CBE-CCA secure assuming that the decision 3-DDH problem is hard in \( G \), \( DEM \) is IND-DEM-CCA secure, and hash functions \( H_1 \) and \( H_2 \) are collision resistant.

This theorem can be proved by combining the following Lemma 2 and Lemma 3.

**Lemma 2.** If \( A \) is a Type-I adversary against our CBE scheme that runs in time \( \tau \), makes at most \( q_c \) queries to the oracle \( \mathcal{O}_{\text{GenerateCertificate}} \) and \( q_D \) queries to the oracle \( \mathcal{O}_{\text{StrongDecrypt}} \), then there exist

- an adversary \( B_1 \) against the collision resistance of the hash function \( H_1 \) that runs in time \( O(\tau) \) and has advantage \( Adv_{CR}^{H_1}(k) \),
- an adversary \( B_2 \) against the collision resistance of the hash function \( H_2 \) that runs in time \( O(\tau) \) and has advantage \( Adv_{CR}^{H_2}(k) \),
- an adversary \( B_3 \) against the decision 3-DDH problem that runs in time \( O(\tau + \epsilon^2 \ln(\epsilon') \lambda^{-1} \ln(\lambda^{-1})) \) and has advantage \( Adv_{3-DDH}^{H_3}(k) \), and
- an adversary \( B_4 \) against the IND-DEM-CCA security of \( DEM \) that runs in time \( O(\tau + \epsilon^2 \ln(\epsilon') \lambda^{-1} \ln(\lambda^{-1})) \) and has advantage \( Adv_{IND-DEM-CCA}^{H_4}(k) \)

such that \( A \)'s advantage is bounded by

\[
Adv_{IND-CBE-CCA}^A(k) \leq Adv_{CR}^{H_1}(k) + Adv_{CR}^{H_2}(k) + 8(q_c + q_D)(n + 1)( Adv_{3-DDH}^{H_3}(k) + Adv_{IND-DEM-CCA}^{H_4}(k) ),
\]

where \( \epsilon = Adv_{3-DDH}^{H_3}(k) \) and \( \lambda = \frac{1}{8(n+1)(q_c + q_D)} \).

**Proof.** The proof of this lemma proceeds by a sequence of seven games. All games involve \( A \) who attempts to guess a random bit \( b \) and eventually outputs a guess \( b' \). For all \( i \), we let \( X_i \) be the event that \( b' = b \) in the Game \( i \) and \( Adv_i \) denote \( A \)'s advantage in the
Game $i$. Then, $\text{Adv}_i = |\Pr[X_i] - 1/2|$. Let $E$ be an event that can occur during the execution of the adversary and is independent of $X_i$ (i.e. $\Pr[X_i|E] = \Pr[X_i]$). We first show that if Game $i+1$ is identical to Game $i$ unless $E$ occurs, then $\text{Adv}_{i+1}$ is negligible if and only if $\text{Adv}_i$ is negligible for $i \geq 1$. Obviously, if $E$ does not occur, $\mathcal{A}$ will output the same bit that it did in Game $i$ (i.e. $\Pr[X_i|\neg E] = \Pr[X_i] = \Pr[X_i]$); otherwise, it will output a random bit (i.e. $\Pr[X_i|\neg E] = 1/2$). Then, we get

$$\Pr[X_{i+1}]\leq |\Pr[X_i|E] - 1/2| + |\Pr[X_i|\neg E] - 1/2| = |\Pr[E]/2 + \Pr[X_i|\neg E] - \Pr[E] - 1/2| = |\Pr[E]/2|.$$ Therefore, we have

$$\text{Adv}_{i+1} = \Pr[\neg E]\text{Adv}_i. \quad (1)$$

We now start the game hopping.

**Game 1.** This game is identical to the original attack game IND-CBE-CCA-I played by $\mathcal{A}$. Therefore, we have $\text{Adv}_1 = \text{Adv}_{\text{IND-CBE-CCA-I}}(k) = |\Pr[X_1] - 1/2|$.

**Game 2.** This game is identical to Game 1 except that the game is stopped if either one of the following two events happens:

- **Event 1**, denoted by $E_1$, is that $\mathcal{A}$ queries $O^{\text{GenerateCertificate}}$ on an identity $id$ such that $(id, PK_{id}) \neq (id', PK_{id'})$ and $H(id, PK_{id}) = H(id', PK_{id'})$, where $id'$ is the challenge identity chosen by $\mathcal{A}$ and $PK_{id'}$ is the public key associated with $id'$.
- **Event 2**, denoted by $E_2$, is that $\mathcal{A}$ queries $O^{\text{StrongDecrypt}}$ on $(id, C = (C_1, C_2, C_3))$ such that $C_i = C_i'$ and $H_2(C_i) = H_2(C_i')$, where $C_i'$ is the first component of the challenge ciphertext $C' = (C_1', C_2', C_3')$.

From Lemma 1, we have $\Pr[X_1] - \Pr[X_2] \leq \Pr[E_1 \lor E_2]$. Clearly, $E_1$ implies a collision for $H_1$ and $E_2$ implies a collision for $H_2$. Hence, if $E_1 \lor E_2$ occurs, then there must exist an algorithm $\mathcal{B}_1$ against the collision resistance of $H_1$ or an algorithm $\mathcal{B}_2$ against the collision resistance of $H_2$ such that $\Pr[E_1 \lor E_2] \leq \text{Adv}_1^{CR}(k) + \text{Adv}_2^{CR}(k)$.

Therefore, we have $|\Pr[X_1] - \Pr[X_2]| \leq \text{Adv}_1^{CR}(k) + \text{Adv}_2^{CR}(k)$. In addition, it is easy to see that the time complexities of the algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ are both bounded by $O(r)$.

**Game 3.** This game is identical to Game 2 except that some values of the public parameters are replaced. The challenger chooses three random values $a, b, y \in Z_p^*$ to set $g_1 = g^a$, $g_2 = g^b$ and $g_3 = g^y$. Let $m = 4q$, where $q = q_c + q_d$ is the total number of certificate and decryption queries that $\mathcal{A}$ makes. The challenger randomly chooses $x, y, x_1, x_2, \ldots, x_n \in [0, m-1]$, $y', y_1, y_2, \ldots, y_m \in Z_p$, $\omega \in [0, n]$, and then sets $u'$ and the vector $U = (u_1, u_2, \ldots, u_m)$ to be $u' = g_2^{\omega - x_m} g^{x_n}$, $u_i = g_2^{x_i} g^{y_1}$ for $1 \leq i \leq n$. Obviously, the replaced values are distributed identically to the corresponding values in Game 2. Therefore, we have $\text{Adv}_3 = \text{Adv}_3$.

**Game 4.** For simplicity, we define two functions from the values of Game 3 as follows.
\begin{align}
J(\lambda) &= (p - \omega m) + x + \sum_{i=1}^{n} \lambda_i x_i, \\
K(\lambda) &= y + \sum_{i=1}^{n} \lambda_i y_i,
\end{align}

where $\lambda = \lambda_1, \lambda_2, \ldots, \lambda_n$ is a $n$-bit string. Then, we have \( F(\lambda) = u_1 u_2^{\lambda_2} \cdots u_n^{\lambda_n} \).

Game 4 is identical to Game 3 except that the game is stopped if $A$ queries \( \mathcal{O}_{\text{GenerateCertificate}} \) or \( \mathcal{O}_{\text{StrongDecrypt}} \) on an identity $id$ such that $J(ID) = 0$ or $A$ wants to be challenged on an identity $id'$ such that $J(ID') \neq 0$, where $ID = H_1(id, PK_0)$ and $ID' = H_1(id', PK_{id'})$. Let $E_3$ be the event that the game is stopped, then $\Pr[-E_3] = \Pr[A, J(ID) \neq 0 \mod p \wedge J(ID') = 0 \mod p]$ where $ID_i = H_1(id_i, PK_{id_i})$. Due to Waters' proof in \cite{Waters}, $\Pr[-E_3] \geq \frac{1}{8(q_c + q_d)(n+1)}$. By Eq. (1), we have $Adv_4 \geq \frac{1}{8(q_c + q_d)(n+1)} Adv_3$.

**Game 5.** In this game, we change the generation of the public parameter $g_1$ and also change the way that the certificate queries and the decryption queries are answered. We set $g_1 = g^a$ to be a random element of $G$ such that $a$ is unknown to the challenger. Now, the challenger answers $A$'s queries to \( \mathcal{O}_{\text{GenerateCertificate}} \) and \( \mathcal{O}_{\text{StrongDecrypt}} \) as follows:

- **$\mathcal{O}_{\text{GenerateCertificate}}$:** On input an identity $id$, the challenger picks a random $s \in \mathbb{Z}_p^*$ and returns $Cert_id = (g_1^{\frac{K(ID)}{ID}} F(ID)^s, g_1^{\frac{1}{ID}} g^s, g_1^{\frac{1}{ID}} g^s)$. If let $s = s - \frac{a}{J(ID)}$, then it is not difficult to deduce that $Cert_id = (g_1^{\frac{K(ID)}{ID}} F(ID)^s, g_1^{\frac{1}{ID}} g^s, g_1^{\frac{1}{ID}} g^s)$. Obviously, it is a valid certificate for the identity $id$.

- **$\mathcal{O}_{\text{StrongDecrypt}}$:** On input an identity $id$ and a ciphertext $C = (C_1, C_2, C_3)$, the challenger first computes $K = e(PK_0^{(2)}, C_2^{\frac{K(ID)}{ID}} / C_1^{\frac{1}{ID} + t})$ where $t = H_2(C_1)$ and $PK_0^{(2)}$ is the second component of the current public key $PK_0$ associated with $id$. It then computes \( DEM.\text{Decrypt}(C_3, K) \) and returns the result. If $C = (C_1, C_2, C_3)$ is a valid ciphertext encrypted using $PK_0$, then we have $C_1 = g^a$ and $C_2 = (g_2^{\frac{K(ID)}{ID}} g^{K(ID)} g_3^s)^r$ for some random $r \in \mathbb{Z}_p^*$. It is easy to deduce that $K = e(PK_0^{(2)}, C_3) g^s$ which is the correct key used to generate the third component $C_3$ of the ciphertext $C$. From the above construction, we can see that the challenger is able to correctly answer the decryption query regardless of whether $PK_0$ is the original public key or not.

We observe that the challenger can correctly answer $A$'s queries to $\mathcal{O}_{\text{GenerateCertificate}}$ and $\mathcal{O}_{\text{StrongDecrypt}}$ as in Game 4. This implies $Adv_4 = Adv_3$.

**Game 6.** This game is identical to Game 5 except that we alter the generation of the
challenge ciphertext. The challenger introduces a new variable \( c \in \mathbb{Z}_p^* \) to set \( C^*_1 = g^c \).

Let \( PK'_{id'} = (PK^{(1)}_{id'}, PK^{(2)}_{id'}) \) be the current public key associated with the challenge identity \( id' \). For a random bit \( \beta \in \{0, 1\} \), it computes \( C^*_1 = (g^\beta)^{K_{id'}+\tau'} \) and \( C^*_2 = DEM.Encrypt(M_{\beta}, K^*) \), where \( ID^* = H_{i}(id', PK'_{id'}) \), \( \tau' = H_2(C^*_1) \) and \( K^* = e(PK^{(1)}_{id'}, g^\text{adv}) \).

It is easy to deduce that \( K^* = e(PK^{(1)}_{id'}, g^\text{adv}) \). Clearly, \( Adv_\beta = Adv_\beta^{} \).

**Game 7.** We again alter the challenge phase. This time, the challenger forgets the values \( b, c \) and simply retains \( g^{b} \) and \( g^{c} \). The challenge ciphertext is constructed as in Game 6 but using a random value \( T \) from \( G \) to compute \( K^* = e(PK^{(1)}_{id'}, T) \). Obviously, Game 7 and Game 6 are equal unless there exists an algorithm \( \beta_1 \) that distinguishes \( T = g^{obc} \) from random. Therefore, we have \( \vert Pr[X_1] - Pr[X_2] \vert \leq Adv_{\beta_1}^{3-DDH}(k) \). In Game 7, \( C^*_1 \) and \( C^*_2 \) are completely independent from the bit \( \beta \) since \( K^* = e(PK^{(1)}_{id'}, T) \) is only a random value of \( G_r \). The adversary in this game essentially carries out a chosen-ciphertext attack on \( DEM \). Therefore, there exists an algorithm \( \beta_2 \) against the IND-DEM-CCA security of \( DEM \) such that \( Adv_{\beta_2} = \vert Pr[X_1] - 1/2 \vert \leq Adv_{\beta_1}^{IND-DEM-CCA}(k) \). In addition, due to Waters’ proof in [20], the time complexities of the algorithms \( \beta_1 \) and \( \beta_2 \) are both bounded by \( O(n + 2^{\lambda}(\lambda)(\log \lambda)) \).

This completes the game hopping. We now summarize the various equalities and inequalities arising in above game hopping steps into a bound on the advantage of the adversary \( \beta \) in the game IND-CBE-CCA-I.

In Game 2 ~ Game 4, we have \( \vert Pr[X_1] - Pr[X_2] \vert \leq Adv_{\beta_2}^{CR}(k) + Adv_{\beta_2}^{CR}(k) \), \( Adv_\beta = Adv_3 \) and \( Adv_1 \leq 8(q_C + q_D)(n + 1)Adv_2 \). Since \( Adv_1 = \vert Pr[X_1] - 1/2 \vert \), we get

\[
Adv_1 \leq \vert Pr[X_1] - Pr[X_2] \vert + \vert Pr[X_1] - Pr[X_3] \vert + \vert Pr[X_3] - 1/2 \vert \\
\leq Adv_{\beta_2}^{CR}(k) + Adv_{\beta_2}^{CR}(k) + 8(q_C + q_D)(n + 1)Adv_2,
\]

(4)

In Game 5 ~ Game 7, we also have \( Adv_4 = Adv_3, Adv_5 = Adv_6, \vert Pr[X_1] - Pr[X_2] \vert \leq Adv_{\beta_2}^{3-DDH}(k) \) and \( Adv_7 = \vert Pr[X_1] - 1/2 \vert \leq Adv_{\beta_1}^{IND-DEM-CCA}(k) \). Then we get

\[
Adv_4 = Adv_5 = Adv_6 = \vert Pr[X_1] - 1/2 \vert \\
\leq \vert Pr[X_1] - Pr[X_2] \vert + \vert Pr[X_3] - 1/2 \vert \\
\leq Adv_{\beta_2}^{3-DDH}(k) + Adv_{\beta_1}^{IND-DEM-CCA}(k),
\]

(5)

Combining (4) and (5), we finally obtain \( Adv_{\beta}^{IND-CBE-CCA-1}(k) = Adv_1 \leq Adv_{\beta_2}^{CR}(k) + Adv_{\beta_2}^{CR}(k) + 8(q_C + q_D)(n + 1)(Adv_{\beta_2}^{3-DDH}(k) + Adv_{\beta_1}^{IND-DEM-CCA}(k)). \) □

**Lemma 3.** If \( \mathcal{A} \) is a Type-II adversary against our CBE scheme that runs in time \( t \), makes at most \( q_D \) queries to the oracle \( O_{\text{dep}} \), then there exist

- an adversary \( \beta_1 \) against the collision resistance of the hash function \( H_2 \) that runs in time \( O(\tau) \) and has advantage \( Adv_{\beta_2}^{CR}(k) \),

\[
\quad
\]
an adversary \( \mathcal{B}_2 \) against the decisional 3-DDH problem that runs in time \( O(\tau + \varepsilon^2 \log(\varepsilon') \lambda^{-1} \log(\lambda^{-1})) \) and has advantage \( \text{Adv}_\mathcal{B}^{3\text{-DDH}}(k) \), and

- an adversary \( \mathcal{B}_3 \) against the IND-DEM-CCA security of **DEM** that runs in time \( O(\tau + \varepsilon^2 \log(\varepsilon') \lambda^{-1} \log(\lambda^{-1})) \) and has advantage \( \text{Adv}_\mathcal{B}^{\text{IND-DEM-CCA}}(k) \)

such that \( \mathcal{A}' \)'s advantage is bounded by

\[
\text{Adv}_\mathcal{A}^{\text{IND-CBE-CCA-II}}(k) \leq \text{Adv}_\mathcal{A}^{\text{CR}}(k) + 8q_D(n+1)(\text{Adv}_\mathcal{A}^{3\text{-DDH}}(k) + \text{Adv}_\mathcal{A}^{\text{IND-DEM-CCA}}(k)),
\]

where \( \varepsilon = \text{Adv}_\mathcal{A}^{3\text{-DDH}}(k) \) and \( \lambda = \frac{1}{8(n+1)q_D} \).

**Proof.** The proof of this lemma is very similar to that of Lemma 1. For all \( i \), we let \( X_i \) be the event that \( b = b_i \) in the Game \( i \) and \( \text{Adv}_i \) denote \( \mathcal{A} \)'s advantage in the Game \( i \).

**Game 1.** This game is identical to the original game IND-CBE-CCA-II played by \( \mathcal{A} \). Therefore, we have \( \text{Adv}_1 = \text{Adv}_\mathcal{A}^{\text{IND-CBE-CCA-II}}(k) = |\text{Pr}[X_1] - 1/2| \).

**Game 2.** This game is identical to Game 1 except that the game is stopped if \( \mathcal{A} \) queries \( O^{\text{Decrypt}} \) on \( (\text{id}, C = (C_1, C_2, C_3)) \) such that \( C_1 \neq C'_1 \) and \( H_2(C_1) = H_2(C'_1) \), where \( C'_1 \) is the first component of the challenge ciphertext \( C' = (C'_1, C'_2, C'_3) \). We denote this event by \( E_1 \). From Lemma 1, we have \( |\text{Pr}[X_1] - \text{Pr}[X_2]| \leq |\text{Pr}[E_1]| \). Clearly, \( E_1 \) implies a collision for the hash function \( H_2 \). Hence, if \( E \) occurs, then there must exist an algorithm \( \mathcal{B}_1 \) against the collision resistance of \( H_2 \) such that \( \text{Pr}[E_1] \leq \text{Adv}_\mathcal{B}^{\text{CR}}(k) \). Therefore, we have \( |\text{Pr}[X_1] - \text{Pr}[X_2]| \leq \text{Adv}_\mathcal{B}^{\text{CR}}(k) \). In addition, it is easy to see that the time complexity of the algorithms \( \mathcal{B}_1 \) is bounded by \( O(\tau) \).

**Game 3.** In this game, the challenger first picks three random values \( \alpha, \beta, \gamma \in Z_p^* \) to set \( g_1 = g^\alpha, g_2 = g^\beta \) and \( g_3 = g^\gamma \). Let \( m = 4q \), where \( q = q_D \) is the total number of decryption queries that \( \mathcal{A} \) makes. It then generates \( u' \) and \( U = (u_1, u_2, \ldots, u_n) \) as in the proof of Lemma 2. Furthermore, it picks a random value \( b \in Z_p^* \) to set \( PK' = (PK_1', PK_2') = (g^\beta, (g^\gamma)^b) \) as the challenge public key. Obviously, the replaced values are distributed identically to the corresponding values in Game 2. Therefore, we have \( \text{Adv}_2 = \text{Adv}_\mathcal{A} \).

**Game 4.** This game is identical to Game 3 except that it is stopped if \( \mathcal{A} \) queries \( O^{\text{Decrypt}} \) on an identity \( \text{id} \) such that \( J(\text{id}) = 0 \) or \( \mathcal{A} \) wants to be challenged on an identity \( \text{id}^* \) such that \( J(\text{id}^*) \neq 0 \), where \( ID = H_1(\text{id}, PK) \) and \( ID' = H_1(\text{id}^*, PK) \). Let \( E_2 \) be the event that the game is stopped, then \( \text{Pr}[E_2] = \text{Pr}[\bigwedge_{i=1}^q J(\text{id}) 
eq 0 \mod p \wedge J(\text{id}^*) = 0 \mod p] \), where \( ID = H_1(\text{id}, PK) \). As in the proof of Lemma 1, we can deduce that \( \text{Adv}_4 \geq \frac{1}{8q_D(n+1)} \text{Adv}_1 \).

**Game 5.** In this game, the challenger answers \( \mathcal{A} \)'s queries to \( O^{\text{Decrypt}} \) as follows: On
input \((id, C = (C_1, C_2, C_3))\), the challenger first computes \(K = e(PK^*_z, C_{i1}^*/C_{i2}^*/t)\), where \(ID = H_1(id, PK^*)\) and \(t = H_2(C_1)\). It then computes \(\text{DEM} . \text{Decrypt}(C_3, K)\) and returns the result. If \(C = (C_1, C_2, C_3)\) is a valid ciphertext encrypted using the challenge public key \(PK^*\), then we have \(C_1 = g^r\) and \(C_2 = (g^{2(ID)}g^{K(ID)}g')'\) for some random value \(r \in Z_p^*\). It is easy to deduce that \(K = e(PK^*_z, g^r)\) which is the correct key used to generate the third component \(C_3\) of the ciphertext \(C\). Therefore, the challenger is able to correctly answer the decryption query even if it does not know the private key corresponding to the challenge public key \(PK^*\). Since the challenger can correctly answer \(A\)'s queries as in Game 4, we have \(Adv_4 = Adv_v\).

**Game 6.** This game is identical to Game 5 except that we alter the generation of the challenge ciphertext \(C^* = (C_1^*, C_2^*, C_3^*)\). The challenger introduces a new variable \(c \in Z_p^*\) to set \(C_1^* = g^c\). For a random bit \(\beta \in \{0, 1\}\), it computes \(C_2^* = (g^\beta g)^{K(ID)+\beta}^*\) and \(C_3^* = \text{DEM} . \text{Encrypt}(M_\beta, K^*)\), where \(ID^* = H_1(id', PK^*)\), \(\tau^* = H_2(C_1^*)\) and \(K^* = e(g_1, g^{abc})\). It is clear that \(K^* = e(PK^*_z, g^r)\). This implies that \(Adv_5 = Adv_v\).

**Game 7.** We again alter the challenge phase. This time, the challenger forgets the values \(a, b, c\) and simply retains \(g^a, g^b\) and \(g^c\). The challenge ciphertext is constructed as in Game 6 but using a random value \(T\) from \(G\) to compute \(K^* = e(g_1, T)\). It is clear that Game 6 and Game 7 are equal unless there exists an algorithm \(A^*\) that distinguishes \(T = g^{abc}\) from random. Therefore, we have \(|\Pr[X_1] - \Pr[X_2]| \leq Adv_{\beta}^{\text{IND-DH}}(k)\). In Game 7, \(C_1^*\) and \(C_2^*\) are completely independent from the bit \(\beta\) since \(K^* = e(g_1, T)\) is only a random value of \(G\). The adversary in this game essentially carries out a chosen-ciphertext attack on \(\text{DEM}\). Therefore, there exists an algorithm \(A^*\) against the IND-DEM-CCA security of \(\text{DEM}\) such that \(Adv_{\gamma} = |\Pr[X_1] - 1/2| \leq Adv_{\beta}^{\text{IND-DEM-CCA}}(k)\). In addition, due to Waters’ proof in [20], the time complexities of the algorithms \(A^*\) and \(A^*\) are both bounded by \(O(\tau + \epsilon^2 \ln(e^{1/2}) \tilde{\omega} \ln(\tilde{\omega}))\).

This completes the game hopping. We now summarize the various equalities and inequalities arising in above game hopping steps into a bound on the advantage of the adversary \(A\) in the game IND-CBE-CCA-II.

In Game 2 ~ Game 4, we have \(|\Pr[X_1] - \Pr[X_2]| \leq Adv_{\beta}^{\text{CR}}(k), Adv_v = Adv_3\) and \(Adv_5 \leq 8q_{\beta}(n+1)Adv_4\). Since \(Adv_1 = |\Pr[X_1] - 1/2|\), we get

\[
Adv_1 \leq |\Pr[X_1] - \Pr[X_2]| + |Pr[X_2] - \Pr[X_3]| + |\Pr[X_3] - 1/2| \\
\leq Adv_{\beta}^{\text{CR}}(k) + 8q_{\beta}(n+1)Adv_4.
\]

(6)

In Game 5 ~ Game 7, we also have \(Adv_4 = Adv_5, Adv_5 = Adv_6, |Pr[X_6] - \Pr[X_7]| \leq Adv_{\beta}^{\text{IND-DH}}(k)\) and \(Adv_7 = |Pr[X_6] - 1/2| \leq Adv_{\beta}^{\text{IND-DEM-CCA}}(k)\). Then we get

\[
Adv_4 = Adv_5 = Adv_6 = |Pr[X_6] - 1/2| \\
\leq |Pr[X_6] - Pr[X_7]| + |Pr[X_7] - 1/2|
\]
\[ \leq \text{Adv}^{\text{3-DDH}}_{A}(k) + \text{Adv}^{\text{IND-DEM-CCA}}_{A}(k). \]  

Combining (6) and (7), we finally obtain \( \text{Adv}^{\text{IND-CBE-CCA-II}}_{A}(k) = \text{Adv}^{\text{CR}}_{A}(k) + 8q_{t}(r+1)(\text{Adv}^{\text{3-DDH}}_{A}(k) + \text{Adv}^{\text{IND-DEM-CCA}}_{A}(k)). \)

\[ \square \]

4.2 Comparison

In this subsection, we make a comparison of our scheme and the previous CBE schemes. We compare the schemes on computation efficiency, communication overhead and some security properties. The details of the compared CBE schemes are listed in Table 2 and Table 3. Note that we do not list all known random-oracle based CBE schemes but some secure and representative ones.

### Table 2. Security properties of the CBE schemes.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Standard model?</th>
<th>Computational assumption(s)</th>
<th>Considering key replacement attack?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>√</td>
<td>3-DDH (=DBDH)</td>
<td>√</td>
</tr>
<tr>
<td>[6]</td>
<td>√</td>
<td>DBDH</td>
<td>×</td>
</tr>
<tr>
<td>[7]</td>
<td>√</td>
<td>DBDH</td>
<td>×</td>
</tr>
<tr>
<td>[8]</td>
<td>√</td>
<td>(q\text{-ABDHE+DBDH})</td>
<td>×</td>
</tr>
<tr>
<td>[9]</td>
<td>×</td>
<td>(p\text{-BDHI+1-BDHI})</td>
<td>×</td>
</tr>
</tbody>
</table>

### Table 3. Performance of the CBE schemes.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Encryption cost</th>
<th>Decryption cost</th>
<th>Ciphertext overhead</th>
<th>Public key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>1Exp+4Exp+2H+E</td>
<td>2P+3Exp+H+D</td>
<td>2(G)</td>
<td>2G</td>
</tr>
<tr>
<td>[6]</td>
<td>2Exp+5Exp+2H+5Mac</td>
<td>3P+3Exp+2H+R+Mac</td>
<td>5(G)+dec+</td>
<td></td>
</tr>
<tr>
<td>[7]</td>
<td>2Exp+5Exp+2H+Sig</td>
<td>3P+3Exp+H+Vfy</td>
<td>3(G)+(</td>
<td>\text{vk}</td>
</tr>
<tr>
<td>[8]</td>
<td>8Exp+2Exp+2H</td>
<td>2P+2Exp+Exp+H</td>
<td>2(G)+</td>
<td>G</td>
</tr>
<tr>
<td>[9]</td>
<td>2Exp+2Exp+1Exp+5H</td>
<td>1P+1Exp+3H</td>
<td>(G)+f</td>
<td>(G)</td>
</tr>
</tbody>
</table>

In the computation cost comparison, we consider four atomic operations: pairing, (long) exponentiation in \(G_{F}\), (short) exponentiation in \(G\) and hash. For simplicity, we denote these operations by \(P\), \(\text{Exp}_{l}\), \(\text{Exp}_{s}\) and \(H\) respectively. In addition, we denote the symmetric encryption and decryption algorithms by \(E\) and \(D\) respectively, the encapsulation and decapsulation algorithms of an encapsulation scheme by \(S\) and \(R\) respectively, the message authentication code algorithm by \(\text{Mac}\), and the one-time signature signing and verification algorithms by \(\text{Sig}\) and \(\text{Vfy}\) respectively. We notice that the costs of some symmetric cryptographic operations (such as hash function, message authentication code, and encapsulation and decapsulation operations, etc.) are usually ignored, as these operations can be implemented efficiently. For our scheme, computing Waters hash \(F(\lambda) = u \prod_{i=1}^{n} u_{i}^{\lambda}\) requires computing \(n/2\) products in \(G\) on the average, and it can be seen as a single exponentiation in \(G\). Therefore, computing \(F(ID)^{r}\) for a random \(r\) can be counted as two exponentiations in \(G\). Once \(F(ID)\) has been precomputed,
computing $F(ID)^r$ needs only one exponentiation in $G$. In the communication cost comparison, ciphertext overhead represents the difference between the ciphertext length and the message length, and the public key size is measured in terms of the number of group elements of the public key. We denote a commitment string and a de-commitment string of an encapsulation scheme by $com$ and $dec$ respectively, a message authentication code by $mac$, a verification key and a signature of a one-time signature scheme by $vk$ and $\sigma$ respectively, and the bit-length of a string $X$ or an element in a group $X$ by $|X|$. In [3] and [9], $l$ should be at least 160 in order to obtain a reasonable security.

In pairing-based cryptography efficiency always depends on the chosen curve. Boyen [26] computes estimated relative timings for all atomic asymmetric operations (exponentiations and pairings) and representation sizes for group elements when instantiated in super-singular curves with 80 bits security (SS/80) and MNT curves with 80 bits security (MNT/80). In Table 4, we recall the data from [26].

Table 4. Timings needed to perform atomic operations and representation of group elements in bits.

<table>
<thead>
<tr>
<th>Curves</th>
<th>Relative timings (1 unit = 1 exp. in $G$)</th>
<th>Representation sizes (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exp in $G$</td>
<td>Exp in $G_T$</td>
</tr>
<tr>
<td>MNT/80</td>
<td>1</td>
<td>36</td>
</tr>
<tr>
<td>SS/80</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5 gives a much clearer comparison between our scheme and the previous standard-model CBE schemes. Note that the costs of the pairings and the exponentiations in $G_T$ are measured by the exponentiations in $G$ according to the data in Table 4. In addition, as the hash operation is much more efficient than the exponentiation in $G$, the costs of the hash operations in all compared schemes are ignored.

Table 5. Performance comparison of the standard-model CBE schemes.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Encryption cost</th>
<th>Decryption cost</th>
<th>Ciphertext overhead</th>
<th>Public key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>$40Exp_s+E$</td>
<td>$303Exp_s+D$</td>
<td>342</td>
<td>342</td>
</tr>
<tr>
<td>SS/80</td>
<td>$8Exp_s+E$</td>
<td>$43Exp_s+D$</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>[6]</td>
<td>$77Exp_s+5*Mac$</td>
<td>$453Exp_s+4*Mac$</td>
<td>513+</td>
<td>dec</td>
</tr>
<tr>
<td>SS/80</td>
<td>$13Exp_s+3*Mac$</td>
<td>$63Exp_s+3*Mac$</td>
<td>1536+</td>
<td>dec</td>
</tr>
<tr>
<td>[7]</td>
<td>$77Exp_s+5*Sig$</td>
<td>$453Exp_s+5*Fv$</td>
<td>513+</td>
<td>vk</td>
</tr>
<tr>
<td>SS/80</td>
<td>$13Exp_s+3*Sig$</td>
<td>$63Exp_s+3*Fv$</td>
<td>1536+</td>
<td>vk</td>
</tr>
<tr>
<td>[8]</td>
<td>$290Exp_s$</td>
<td>$373Exp_s$</td>
<td>2223</td>
<td>1026</td>
</tr>
<tr>
<td>SS/80</td>
<td>$34Exp_s$</td>
<td>$49Exp_s$</td>
<td>2560</td>
<td>3072</td>
</tr>
</tbody>
</table>

From Table 5, we conclude that our scheme outperforms all the previous standard-model CBE schemes. Interestingly, it seems to also offer competitive computation efficiency in comparison with the random-oracle based CBE scheme proposed by Gentry [3]. Furthermore, the message spaces of all the previous standard-model CBE schemes are restricted in $G_T$, which is about 1024 bits (when instantiated in super-singular curves with 80 bits security). Thus, when encrypting a
large message (e.g. $10^4$ bits), these schemes have to split the message into many parts and encrypt it part by part. It results in many times increase of both computation cost and ciphertext size, and maybe security reduction as well. Due to the use of the KEM-DEM hybrid encryption framework, our scheme does not suffer from the message length restriction. Therefore, it can be used to encrypt arbitrary-length messages without any efficiency reduction.

5. CONCLUSIONS

In this paper, we have proposed a new CBE scheme without random oracles. We have proved in the standard model that the proposed scheme is secure against both public key replacement and adaptive chosen-ciphertext attacks under the hardness of the 3-DDH problem. Performance analysis shows that the proposed scheme performs all the previous standard-model CBE schemes in the literature. To our best knowledge, it is the first encryption scheme in the certificate-based setting that has been explicitly proven secure against key replacement attacks in the standard model.

REFERENCES


Yang Lu (陆阳) received the Ph.D. degree from PLA University of Science and Technology in 2009. He has been working in HoHai University from 2003. Currently, he is an Assistant Professor in College of Computer and Information Engieering. He has published more than 30 papers in international conferences and journals. His research interest includes information security and cryptography.
Jiguó Li (李继国) received the Ph.D. degree from Harbin Institute of Technology in 2003. He has been working in HoHai University from 2003. Currently, he is a Professor in College of Computer and Information Engineering. His major research interests include information security and cryptography, network security, wireless security and trusted computing etc. He has published more than 70 scientific papers and two books. He has served as a PC member of several international conferences and the reviewer of some international journals and conferences.