Identity-Based Encryption with Cloud Revocation Authority and Its Applications

Yuh-Min Tseng, Tung-Tso Tsai, Sen-Shan Huang, and Chung-Peng Huang

Abstract—Identity-based encryption (IBE) is a public key cryptosystem and eliminates the demands of public key infrastructure (PKI) and certificate administration in conventional public key settings. Due to the absence of PKI, the revocation problem is a critical issue in IBE settings. Several revocable IBE schemes have been proposed regarding this issue. Quite recently, by embedding an outsourcing computation technique into IBE, Li *et al*. proposed a revocable IBE scheme with a key-update cloud service provider (KU-CSP). However, their scheme has two shortcomings. One is that the computation and communication costs are higher than previous revocable IBE schemes. The other shortcoming is lack of scalability in the sense that the KU-CSP must keep a secret value for each user. In the article, we propose a new revocable IBE scheme with a cloud revocation authority (CRA) to solve the two shortcomings, namely, the performance is significantly improved and the CRA holds only a system secret for all the users. For security analysis, we demonstrate that the proposed scheme is semantically secure under the decisional bilinear Diffie-Hellman (DBDH) assumption. Finally, we extend the proposed revocable IBE scheme to present a CRA-aided authentication scheme with period-limited privileges for managing a large number of various cloud services.

Index Terms—Encryption, authentication, cloud computing, outsourcing computation, revocation authority.

1 Introduction

IDENTITY (ID)-based public key system (ID-PKS) [1], [2] is an attractive alternative for public key cryptography. ID-PKS setting eliminates the demands of public key infrastructure (PKI) and certificate administration in conventional public key settings. An ID-PKS setting consists of users and a trusted third party (i.e. private key generator, PKG). The PKG is responsible to generate each user's private key by using the associated ID information (e.g. e-mail address, name or social security number). Therefore, no certificate and PKI are required in the associated cryptographic mechanisms under ID-PKS settings. In such a case, ID-based encryption (IBE) allows a sender to encrypt message directly by using a receiver's ID without checking the validation of public key certificate. Accordingly, the receiver uses the private key associated with her/his ID to decrypt such ciphertext. Since a public key setting has to provide a user revocation mechanism, the research issue on how to revoke misbehaving/compromised users in an ID-PKS setting is naturally raised.

In conventional public key settings, certificate revocation list (CRL) [3] is a well-known revocation approach. In the CRL approach, if a party receives a public key and its associated certificate, she/he first validates them and then looks up the CRL to ensure that the public key has not been revoked. In such a case, the procedure requires the online assistance under PKI so that it will incur communication bottleneck. To improve the performance, several efficient revocation mechanisms [4], [5], [6], [7], [8] for conventional public key settings have been well studied for PKI. Indeed, researchers also pay attention to the revocation issue of ID-PKS settings. Several revocable IBE schemes have been proposed regarding the revocation mechanisms in ID-PKS settings.

1.1 Related Work

In 2001, Boneh and Franklin [2] proposed the first practical IBE scheme from the Weil pairing and suggested a simple revocation method in which each non-revoked user receives a new private key generated by the PKG periodically. A period can be set as a day, a week, a month, etc. A sender uses a designated receiver's ID and current period to encrypt messages while the designated receiver decrypts the ciphertext using the current private key. Hence, it is necessary for the users to update new private keys periodically. To revoke a user, the PKG simply stops providing the new private key for the user. It is obvious that a secure channel must be established between the PKG and each user to transmit the new private key and this would result in heavy load for the PKG.

In order to alleviate the load of the PKG in Boneh and Franklin's scheme, Boneh *et al.* [9] proposed another revocation method, called immediate revocation. Immediate revocation method employs a designated semi-trusted and online authority (i.e. mediator) to mitigate the management load of the PKG and assist users to decrypt ciphertext [10], [11], [12], [13]. In such a case, the online mediator must hold shares of all the users' private keys. Since the decryption operation must involve both parties, neither the user nor the online mediator can cheat one another. When a user was

- *Y.-M. Tseng is with the Department of Mathematics, National Changhua University of Education, Chang-Hua City 500, Taiwan, corresponding authors e-mail: (ymtseng@cc.ncue.edu.tw).*
- *T.-T. Tsai is with the HON HAL Technology Group, Taiwan.*
- *S.-S. Huang and C.-P. Huang are with the Department of Mathematics, National Changhua University of Education, Chang-Hua City 500, Taiwan.*

revoked, the online mediator is instructed to stop assisting the user. However, the online mediator must help users to decrypt each ciphertext so that it becomes a bottleneck for such schemes as the number of users grows enormously.

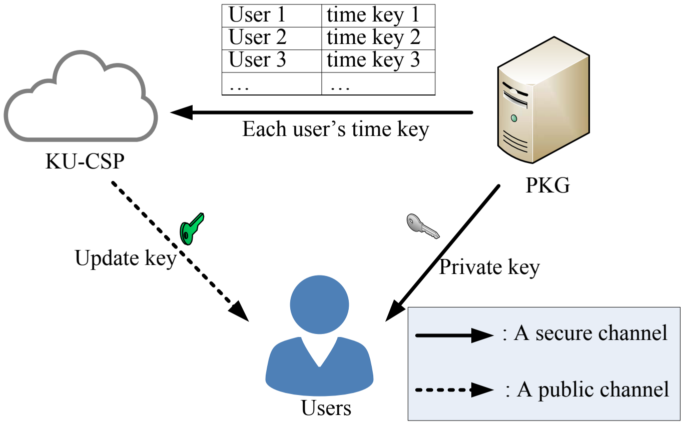
On the other hand, in Boneh and Franklin's revocation method [2], all the users must periodically update new private keys sent by the PKG. As the number of users increases, the load of key updates becomes a bottleneck for the PKG. In 2008, Boldyreva *et al*. [14] proposed a revocable IBE scheme to improve the key update efficiency. Their revocable IBE scheme is based on the concept of the Fuzzy IBE [35] and adopts the complete subtree method to decrease the number of key updates from linear to logarithmic in the number of users. Indeed, by binary tree data structure of users, the scheme efficiently alleviates the key-update load of the PKG. Furthermore, Libert and Vergnaud [16] improved the security of Boldyreva *et al*.'s revocable IBE scheme by presenting an adaptive-ID secure scheme. Nevertheless, Boldyreva *et al*.'s scheme still results in several problems: (1) Each user's private key size is 3log n points in an elliptic curve, where n is the number of leaf nodes (users) in the binary tree. (2) The scheme also results in enormous computation workload for encryption and decryption procedures. (3) It is enormous load for PKG to maintain the binary tree with a large amount of users.

Moreover, Seo and Emura [17] refined the security model of Boldyreva *et al*.'s revocable IBE scheme [14] by considering a new threat, called decryption key exposure attacks. Based on the idea of Libert and Vergnaud's scheme [16], they also proposed a revocable IBE scheme with decryption key exposure resistance. In order to reduce the sizes of both private keys and update keys, Park *et al*. [18] proposed a new revocable IBE scheme by using multilinear maps, but the size of the public parameters is dependent to the number of users. For achieving constant the size of the public parameters, Wang *et al*. [19] employed both the dual system encryption methodology [20] and the complete subtree method [14] to propose a new revocable IBE scheme.

Furthermore, Seo and Emura [21] extended the concept of revocable IBE scheme to propose the first revocable HIBE scheme. In Seo and Emura's scheme, for each period, each user generates a secret key by multiplying some of the partial keys, which depends on the partial keys used by ancestors in the hierarchy tree. In such a case, the secret key size of each user increases quadratically in the hierarchy tree wherein a low-level user must know the history of key updates performed by ancestors in the current time period, and it renders the scheme very complex. In 2015, Seo and Emura [22] proposed a new method to construct a novel revocable HIBE scheme with history-free updates. Nevertheless, the mentioned revocable IBE and HIBE schemes above [17], [18], [19], [21], [22] employed the complete subtree method to decrease the number of key updates from linear to logarithmic in the number of users. However, these schemes also suffered from the same disadvantages of Boldyreva *et al*.'s revocable IBE scheme [14] and still used a secure channel to transmit periodic private keys.

In 2012, Tseng and Tsai [23] proposed a new revocable IBE scheme to remove the usage of secure channel between each user and the authority and use a public channel instead to transmit users' periodic private keys. They partition a user's private key into two components, namely, an identity key and a time update key. The identity key is a secret key associated with user's ID, which is sent to the user via a secure channel and remains fixed since being issued. The time update key is a key associated with user's ID and time period, which is changed along with time. The PKG periodically generates current time update keys for non-revoked users and sends them to these users via a public channel. A user is able to decrypt the ciphertext if she/he possesses both the identity key and the legitimate time update key. In other words, to revoke a particular user, the PKG simply stops issuing the new time update key for the user. However, the key-update efficiency is linear in the number of users so that the computation burden of PKG is still enormous.

In 2015, by a cloud-aided service provider, Li *et al*. [24] introduced an outsourcing computation technique into IBE to propose a revocable IBE scheme with a key-update cloud service provider (KU-CSP). They shifts the key-update procedures to a KU-CSP to alleviate the load of PKG. Li *et al*. also used the similar technique adopted in Tseng and Tsai's scheme [23], which partitions a user's private key into an identity key and a time update key. The PKG sends a user the corresponding identity key via a secure channel. Meanwhile, the PKG must generate a random secret value (time key) for each user and send it to the KU-CSP. Then the KU-CSP generates the current time update key of a user by using the associated time key and sends it to the user via a public channel. To revoke a user, the PKG just asks the KU-CSP to stop issuing the new time update key of the user. Their system model is depicted in Fig. 1. However, their scheme has two shortcomings. One is that the computation and communication costs are higher than previous revocable IBE schemes [2], [23]. The other shortcoming is un-scalability in the sense that the KU-CSP must keep a time key for each user so that it will incur the management load.



User 1	time key 1
User 2	time key 2
User 3	time key 3
…	…

Fig. 1: Li *et al*.'s system model

1.2 Our Contributions

In order to solve both the un-scalability and the inefficiency in Li *et al*.'s scheme [24], we will propose a new revocable IBE scheme with cloud revocation authority (CRA). The proposed scheme possesses the advantages of both Tseng

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2016.2541138, IEEE Transactions on Cloud Computing

TABLE 1: Comparisons of previous revocable IBE and HIBE schemes and ours

	Subtree-based IBE and HIBE schemes [14], [16], [17], [18], [19], [21], [22]	Tseng-Tsai scheme [23]	Li *et al.*'s scheme [24]	Our scheme
Key update channel	Secure channel	Public channel	Public channel	Public channel
The size of each user's private key	$O(\log n)$	$O(1)$	$O(1)$	$O(1)$
Total key update load	$O(\log n)$	$O(n)$	$O(n)$	$O(n)$
Outsourced computation of authority	No	No	Yes	Yes
Workload of the PKG	Medium	High	Low	Low
Scalability of authority	No support	No support	Un-scalability	Yes

and Tsai's revocable IBE scheme [23] and Li *et al.*'s scheme [24]. In particular, each user's private key still consists of an identity key and a time update key. We introduce a cloud revocation authority (CRA) to replace the role of the KU-CSP in Li *et al.*'s scheme. The CRA only needs to hold a random secret value (master time key) for all the users without affecting the security of revocable IBE scheme. The CRA uses the master time key to generate the current time update key periodically for each non-revoked user and sends it to the user via a public channel. It is evident that our scheme solves the un-scalability problem of the KU-CSP. Our system model is depicted in Fig. 2.

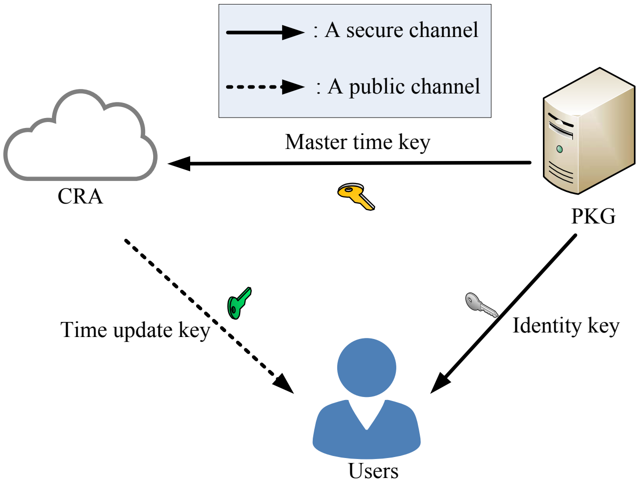


Fig. 2: System model for revocable IBE scheme with CRA

In this article, we first present the framework of our revocable IBE scheme with CRA and define its security notions to model possible threats and attacks. Accordingly, a new revocable IBE scheme with CRA is proposed. As the adversary model presented in [23], [24], it consists of two adversaries, namely, an inside adversary (or a revoked user) and an outside adversary. For security analysis, we formally demonstrate that our scheme is semantically secure against adaptive-ID and chosen-ciphertext attacks (CCA) in the random oracle model under the bilinear decision Diffie-Hellman problem [2]. Finally, based on the proposed revocable IBE scheme with CRA, we construct a CRA-aided authentication scheme with period-limited privileges for managing a large number of various cloud services.

To demonstrate the merits of our scheme, Table 1 lists the comparisons among subtree-based IBE schemes [14], [16], [17], [18], [19], HIBE schemes [21], [22], Tseng-Tsai scheme [23], Li *et al.*'s scheme [24] and ours in terms of the usage of key update channel, the size of each user's private key, key update load, outsourced computation of authority, the workload of the PKG and scalability of authority.

Those subtree-based IBE schemes [14], [16], [17], [18], [19] and HIBE schemes [21], [22] employed the complete subtree method to decrease the number of key updates from linear to logarithmic in the number of users. However, each user's private key size is $O(\log n)$, where n is the number of users. These schemes still used a secure channel to transmit periodic private keys while no other authority shares the responsibility of user revocation. In Tseng and Tsai's revocable IBE scheme [23], both the identity key and time update key are issued by the PKG. In order to alleviate the load of the PKG, Li *et al.* [24] employed a key update cloud service provider (KU-CSP) to share the responsibility of user revocation. In our revocable IBE scheme, we employ a cloud revocation authority (CRA) to perform user revocation. Indeed, the PKG in Li *et al.*'s scheme and ours may also perform the revocation operations. Both the KU-CSP and the CRA are designated to share responsibility for performing user revocation. For scalability, the KU-CSP in Li *et al.*'s scheme must keep n various time keys for n users so that it does not possess scalability and incurs the management load. On the contrast, the CRA in our scheme holds only one master time key for all the users. When the number n of users in the system is very large, the PKG may designate multiple CRAs to share the responsibility of user revocation while each CRA holds only the same master time key. However, in Li et al.'s scheme, each KU-CSP must also keep n time keys. Indeed, cloud computing is a ubiquitous computing environment so that putting multiple CRAs on clouds may provide convenient management of user revocation while reducing the load of the single PKG. The detailed comparisons regarding computation and communication efficiency will be given in Section 6.

1.3 Organization

The remainder of the article is organized as follows. Preliminaries are presented in Section 2. In Section 3, we introduce the system environment, and define the syntax and security model for our revocable IBE scheme with CRA. A concrete construction is presented in Section 4. In Sections 5 and 6, we demonstrate the security analysis and performance analysis of our scheme, respectively. Based on our scheme, two extended cloud computing applications are presented in Section 7. Lastly, we draw a conclusion in Section 8.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2016.2541138, IEEE Transactions on Cloud Computing

2 Preliminaries

2.1 Bilinear Pairings

We first define several notations of bilinear pairings [2], [25] as follows:

- G is an additive cyclic group of a prime order q.
- G_T is a multiplicative cyclic group of the same prime order q.
- P is a generator of G.

We say that $\hat{e} : G \times G \rightarrow G_T$ is an admissible bilinear map if it possesses the following three properties:

(1) Bilinearity: for all $Q, R \in G$ and $a, b \in Z_q^*$, we have $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$.
(2) Non-degeneracy: $\hat{e}(P, P)$ generates G_T.
(3) For practical purposes, $\hat{e}$ has to be computable in an efficient manner.

Note that an admissible bilinear map $\hat{e}$ is symmetric since $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab} = \hat{e}(bP, aP)$.

2.2 Complexity Assumption

The security of our scheme is established under the decisional bilinear Diffie-Hellman (DBDH) assumption [2]. We describe the DBDH problem and define its associated assumption as follows.

DBDH problem. Let G and G_T be two cyclic groups of a large prime order q and P be a generator of G. Let $\hat{e} : G \times G \rightarrow G_T$ be an admissible bilinear map. The DBDH problem in $< G, G_T, \hat{e} >$ is stated as below: given $P, aP, bP, cP \in G$ with unknown $a, b, c \in Z_q^*$ and a random value $T \in G_T$, to decide if $T = \hat{e}(P, P)^{abc}$.

DBDH assumption. We say that the DBDH assumption holds in $< G, G_T, \hat{e} >$ if no polynomial-time algorithm can solve the DBDH problem with non-negligible advantage.

3 System Operations, Framework and Security Notions

For convenience, we first define the following notations.

- α: the master secret key.
- β: the master time key.
- P_{pub}: the system public key $P_{pub} = \alpha \cdot P$.
- C_{pub}: the cloud public key $C_{pub} = \beta \cdot P$.
- ID: the identity of a user, $ID \in \{0, 1\}^*$.
- D_{ID}: the identity key of the user with identity ID.
- i: the period index, where $1 \leq i \leq z$ and z denotes the total number of periods.
- $P_{ID,i}$: the time update key of the user with ID for period i.
- H_0: a hash function $H_0 : \{0, 1\}^* \rightarrow G$.
- H_1: a hash function $H_1 : \{0, 1\}^* \rightarrow G$.
- H_2: a hash function $H_2 : G_T \rightarrow \{0, 1\}^l$, where l is a fixed length.
- H_3: a hash function $H_3 : \{0, 1\}^* \rightarrow \{0, 1\}^l$.

3.1 System Operations

In Fig. 3, we present the system operations of the proposed revocable IBE scheme with CRA. Our system has three roles, namely, a private key generator (PKG), a cloud revocation authority (CRA) and users (senders and receivers). First, the PKG selects a master secret key α, a master time key β and a total number z of periods, and sends the master time key β to the CRA. The PKG uses the master secret key α to compute the identity key D_{ID} of the user with identity ID, and sends the identity key D_{ID} to the user via a secure channel. On the other hand, the CRA is responsible to produce the time update keys for all the non-revoked users by using the master time key β. To do this, at the starting of each period i, the CRA uses the master time key β and a non-revoked user's identity ID to generate the current time update key $P_{ID,i}$, and sends it to the user via a public channel (e.g. e-mail).

When a sender wants to transmit a message M to a receiver with identity ID at period i, the sender produces a ciphertext $C = E(ID, i, M)$ and sends it to the receiver, where E denotes the encryption algorithm of our revocable IBE scheme with CRA. Upon receiving the ciphertext, the receiver uses the identity key D_{ID} and time update key $P_{ID,i}$ to decrypt the ciphertext.

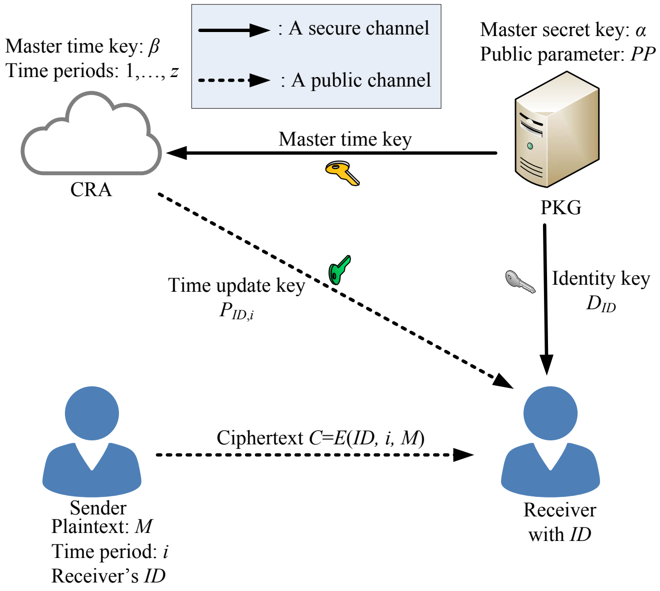


Fig. 3: System operations of revocable IBE scheme with CRA

3.2 Framework

In this section, we present the syntax of revocable IBE schemes with CRA.

Definition 1. A revocable IBE scheme with CRA consists of five algorithms: *system setup*, *identity key extract*, *time key update*, *encryption* and *decryption*.

- *System setup* is a probabilistic algorithm that is run by the PKG. The PKG takes as input two parameters, namely, a secure parameter λ and the total number z of periods, and outputs public parameters PP, a master secret key α and a master time key β. Finally, it sends β to the CRA via a secure channel. PP are made public to all the following algorithms.

- *Identity key extract* is a deterministic algorithm which is run by the PKG that takes as input the master secret key α and a user's identity ID, and outputs the corresponding identity key D_{ID}. Then, the PKG returns D_{ID} to the user via a secure channel.
- *Time key update* is a deterministic algorithm which is run by the CRA. The CRA uses the master time key β, a user's identity ID and a period i to compute the user's time update key $P_{ID,i}$ for period i. Then, the CRA returns the time update key $P_{ID,i}$ to the user via a public channel (e.g. e-mail or public board).
- *Encryption* is probabilistic algorithm that is run by a user (sender). The sender takes as input a message M, a receiver's identity ID and a current period i, and outputs a ciphertext C.
- *Decryption* is a deterministic algorithm which is run by a user (receiver). The receiver takes as input a ciphertext C and the private key pair (D_{ID}, $P_{ID,i}$), and outputs the corresponding plaintext M.

3.3 Security Notions

Here, we give the formal security notions for revocable IBE schemes with CRA. Like the security notions in [23], [24], there are two types of adversaries, namely, Type I adversary A_I (a revoked user) and Type II adversary A_{II} (an outsider or a curious CRA). Two types of adversaries are described as follows.

- *Type I adversary A_I (a revoked user)*. This adversary used to be a legal user with identity ID^* of the system who has been revoked by the CRA at some period i^*. Such an adversary would like to decrypt ciphertexts sent to him/her at period i^* with the assumption that it can obtain the identity key of every user. Meantime, the adversary is able to obtain the time update keys of all the users at arbitrary period, except the target identity ID^* at period i^*.
- *Type II adversary A_{II} (an outsider or a curious CRA)*. Evidently, the CRA can compute the time update keys for all the users at arbitrary period since it owns the master time key β. On the other hand, an outsider also knows all the time update keys published by the CRA via a public channel. Therefore, an adversary of Type II can obtain the identity key of any user, except the target identity ID^*.

Following the security notions of revocable IBE schemes in [23], [24], we define the security notions for revocable IBE schemes with CRA that include two types of the indistinguishability of encryption, namely, under adaptive ID and chosen-plaintext attacks (IND-ID-CPA), and under adaptive ID and chosen-ciphertext attacks (IND-ID-CCA), respectively. Here, we first present two security games to define the IND-ID-CCA attacks for adversaries of Types I and II, respectively.

Game 1 (Type I adversary A_I):

- *System setup*. The challenger B takes a security parameter λ and runs the $System\ setup$ algorithm to obtain a master secret key α, a master time key β and public parameters PP. It forwards PP to the adversary A_I while α and β are kept secret by B.
- *Phase* 1. The adversary A_I is allowed to issue the following queries in an adaptive manner.
    - *Identity key extract query* (ID). When A_I issues such a request along with a user's identity $ID \in \{0,1\}^*$, B runs the *Identity key extract* algorithm to generate the identity key D_{ID} and sends it to A_I.
    - *Time key update query* (ID, i). When A_I issues such a request along with a user's identity $ID \in \{0,1\}^*$ and a period i, B runs the *Time key update* algorithm to generate the time update key $P_{ID,i}$ and responds with it.
    - *Decryption query* (C, ID, i). Upon receiving the query along with a ciphertext C, a user's identity $ID \in \{0,1\}^*$ and a period i, B obtains the private key pair $(D_{ID}, P_{ID,i})$ by issuing the *Identity key extract* query with ID and the *Time key update* query with (ID, i). The challenger B runs the $Decryption$ algorithm to decrypt the ciphertext C and returns the corresponding plaintext M to A_I.
- *Challenge*. A_I sends a plaintext pair (M_0, M_1), a user's identity ID^* and a period i^* to the challenger B. Then B flips a random coin $\gamma \in \{0,1\}$, sets the ciphertext $C^* = E(ID^*, i^*, M_\gamma)$ and returns C^* to A_I. Here, we require that (ID^*, i^*) did not appear in the *Time key update* query of the $Phase$ 1.
- *Phase* 2. A_I may issue further queries as those in the $Phase$ 1. The only restriction is that A_I cannot issue the *Time key update* query with (ID^*, i^*) and the $Decryption$ query with (ID^*, i^*, C^*).
- *Guess*. A_I outputs a guess bit $\gamma' \in \{0,1\}$ and wins the game if $\gamma' = \gamma$.

Game 2 (Type II adversary A_{II}):

- *System setup*. This phase is identical to the *System setup* phase in **Game 1**.
- *Phase* 1. The adversary A_{II} can adaptively issue all the queries in the $Phase$ 1 of **Game 1**.
- *Challenge*. A_{II} sends a plaintext pair (M_0, M_1), a user's identity ID^* and a period i^* to the challenger B. Then B flips a random coin $\gamma \in \{0,1\}$, sets the ciphertext C^*= $E(ID^*, i^*, M_\gamma)$ and returns C^* to A_{II}. Here, we require that ID^* did not appear in the *Identity key extract* query of the $Phase$ 1.
- *Phase* 2. A_{II} may issue further queries as those in the $Phase$ 1. The only restriction is that A_{II} cannot issue the *Identity key extract* query with ID^* and the $Decryption$ query with (ID^*, i^*, C^*).
- *Guess*. A_I outputs a guess bit $\gamma' \in \{0,1\}$ and wins the game if $\gamma' = \gamma$.

In the games above, we refer to such A_I and A_{II} as polynomial-time adversaries. The advantage of an IND-ID-CCA adversary A (A_I or A_{II}) to attack the revocable IBE scheme with CRA is defined by the function $Adv_A(\lambda) = |\Pr[\gamma = \gamma'] - \frac{1}{2}|$, where λ is the security parameter.

Definition 2. *We say that a revocable IBE scheme with CRA is semantically secure against adaptive chosen-ciphertext attacks (IND-ID-CCA) if no probability-polynomial-time (PPT) adversary A has a non-negligible advantage in Games* 1 *or* 2.

For the indistinguishability of encryption under adaptive ID and chosen-plaintext attacks (IND-ID-CPA), two security games are the same as Games 1 and 2, except that an adversary cannot issue the $Decryption$ query.

Definition 3. *We say that an IBE-CRA is semantically secure against an adaptive chosen-plaintext attack (IND-ID-CPA) if no PPT adversary has a non-negligible advantage in Games* 1 *or* 2. *Here, we require that an adversary cannot issue the Decryption query in Phases* 1 *or* 2.

4 The Proposed Revocable IBE Scheme with CRA

Here, we propose an efficient revocable IBE scheme with CRA. The scheme is constructed by using bilinear pairings (Section 2) and consists of five algorithms as the framework defined in Section 3.2.

- *System setup*: A trusted PKG takes as input two parameters, namely, a secure parameter λ and the total number z of periods. The PKG randomly chooses two cyclic groups G and G_T of a prime order $q > 2^\lambda$. Also, it randomly chooses a generator P of G, an admissible bilinear map $\hat{e} : G\times G \to G_T$ and two secret values $\alpha, \beta \in Z_q^*$. The value α is the master secret key used to compute the system public key $P_{pub} = \alpha \cdot P$. The PKG then transmits the master time key β to the CRA via a secure channel. The value β is used to compute the cloud public key $C_{pub} = \beta \cdot P$. The PKG selects three hash functions $H_0, H_1 : \{0,1\}^* \to G$, $H_2 : G_T \to \{0,1\}^l$, and $H_3 : \{0,1\}^* \to \{0,1\}^l$, where l is fixed, and publishes the public parameters $PP =< q, G, G_T, \hat{e}, P, P_{pub}, C_{pub}, H_0, H_1, H_2, H_3 >$.
- *Identity key extract*: Upon receiving the identity $ID \in \{0,1\}^*$ of a user, the PKG uses the master secret key α to compute the corresponding identity key $D_{ID} = \alpha \cdot S_{ID}$, where $S_{ID} = H_0(ID)$. Then, the PKG sends the identity key D_{ID} to the user via a secure channel.
- *Time key update*: To generate the time update key $P_{ID,i}$ at period i for a user with identity $ID \in \{0,1\}^*$, the CRA uses the master time key β to compute the time update key $P_{ID,i} = \beta\cdot T_{ID,i}$, where $T_{ID,i} = H_1(ID, i)$. Finally, the CRA sends the time update key $P_{ID,i}$ to the user via a public channel.
- *Encryption*: To encrypt a message $M \in \{0,1\}^l$ with a receiver's identity ID and a period i, a sender selects a random value $r \in Z_q^*$ and computes $U = r \cdot P$. The sender also computes $V = M \oplus H_2((g_1 \cdot g_2)^r)$, where $g_1 = \hat{e}(S_{ID}, P_{pub})$ and $g_2 = \hat{e}(T_{ID,i}, C_{pub})$. Then, the sender computes $W = H_3(U, V, M, ID, i)$. Finally, the sender sets the ciphertext as $C = (U, V, W)$ and sends it to the receiver.
- *Decryption*: To decrypt a ciphertext $C = (U, V, W)$ with a receiver's identity ID and a period i, the receiver uses his/her identity key D_{ID} and time update key $P_{ID,i}$ to compute the plaintext $M = V \oplus H_2(\hat{e}(D_{ID} + P_{ID,i}, U))$. If $W = H_3(U, V, M, ID, i)$, return M as the plaintextoutput, else return $\perp$.

The correctness of the decryption algorithm follows since

$$\begin{aligned} V{\oplus}H_2(\hat{e}(D_{ID} + P_{ID,i}, U)) & \\ &= M \oplus H_2((g_1 \cdot g_2)^r) \oplus H_2(\hat{e}(D_{ID} + P_{ID,i}, U)) \\ &= M \oplus H_2((g_1 \cdot g_2)^r) \oplus H_2(g_1^r \cdot g_2^r) \\ &= M, \end{aligned}$$

where the penultimate equality is due to the fact that

$$\begin{aligned} H_2(\hat{e}(D_{ID} + P_{ID,i}, U)) & \\ &= H_2(\hat{e}(D_{ID}, U) \cdot \hat{e}(P_{ID,i}, U)) \\ &= H_2(\hat{e}(\alpha \cdot S_{ID}, r \cdot P) \cdot \hat{e}(\beta \cdot T_{ID,i}, r \cdot P)) \\ &= H_2(\hat{e}(S_{ID}, \alpha \cdot P)^r \cdot \hat{e}(T_{ID,i}, \beta \cdot P)^r) \\ &= H_2(\hat{e}(S_{ID}, P_{pub})^r \cdot \hat{e}(T_{ID,i}, C_{pub})^r) \\ &= H_2(g_1^r \cdot g_2^r). \end{aligned}$$

Note that the proposed scheme above will be proved to be an IND-ID-CCA-secure IBE scheme in the next section. Indeed, a simple IND-ID-CPA-secure IBE scheme is obtained by removing W from $C = (U, V, W)$ in the proposed scheme, namely, the ciphertext only consists of $C = (U, V)$. On the contrast, Tseng and Tsai's scheme [23], and Li *et al.*'s scheme [24] are IND-ID-CPA-secure IBE schemes. Their schemes must use the transformation methods in [26], [27] to transform an IND-ID-CPA-secure IBE scheme into an IND-ID-CCA-secure IBE scheme. In such a case, the ciphertexts of their schemes have to add a hash value W in ciphertext as our proposed scheme.

5 Security Analysis

In this section, we present the formal security analysis of our revocable IBE scheme with CRA. Lemmas 1 and 2 are given to demonstrate that our scheme is semantically secure against adversaries of Types I and II, respectively. By Lemmas 1 and 2, we conclude that our scheme possesses the indistinguishability of encryption under adaptive ID and chosen-ciphertext attacks (IND-ID-CCA).

Lemma 1. *In the random oracle model, suppose that there is a Type I adversary A_I with probability ϵ who can break the proposed revocable IBE scheme with CRA in Game* 1*. In the meantime, let q_u and q_d denote, respectively, time key update queries and decryption queries that A_I is allowed to issue. Then we can construct an algorithm B who solves the DBDH problem with probability*

$$\epsilon' \geq \frac{\epsilon}{e(1+q_u)} - \frac{q_d}{q},$$

where e is the base value of the natural logarithm.

Proof. Assume that there is a Type I adversary A_I with probability ϵ who can break the proposed revocable IBE scheme with CRA. We construct an algorithm B to solve the DBDH problem with probability ϵ'. The algorithm B takes as input the DBDH parameters $< q, G, G_T, \hat{e} >$ and a tuple $< P, aP, bP, cP, T >$, where P is a generator of the group G, $a, b, c \in Z_q^*$ are unknown to B and $T \in G_T$. Next, the algorithm B attempts to solve the DBDH problem by deciding if $T = \hat{e}(P, P)^{abc}$. Here, B acts as the challenger and interacts with A_I in Game 1 as follows.

- *System setup*. The challenger B first chooses a random master secret key $\alpha \in Z_q^*$ and sets $P_{pub} = \alpha \cdot P$. Then B provides A_I with PP = $<$ q, G, G_T, $\hat{e}$, P, P_{pub}, C_{pub}, H_0, H_1, H_2, H_3 $>$, where $C_{pub} = aP$. Moreover, H_0, H_1, H_2 and H_3 are random oracles controlled by B defined as below.
  - *H_0-queries*: The challenger B maintains a list L_0 of tuples $< ID, S_{ID}, u >$. Upon receiving the query along with ID, B performs the following steps:
    (1) If ID appears in L_0, then B responds with $H_0(ID) = S_{ID}$.
    (2) If ID does not appear in L_0, B chooses a random value $u \in Z_q^*$ and computes $S_{ID} = u \cdot P$. B adds $< ID, S_{ID}, u >$ in L_0 and returns $H_0(ID) = S_{ID}$ to A_I.
  - *H_1-queries*: The challenger B maintains a list L_1 of tuples $< ID, i, T_{ID,i}, v, coin >$. Upon receiving the query along with (ID, i), B performs the following steps:
    (1) If (ID, i) appears in L_1, then B responds with $H_1(ID, i)$ = $T_{ID,i}$.
    (2) If (ID, i) does not appear in L_1, then B chooses a random value $v \in Z_q^*$. B then flips a random $coin \in \{0, 1\}$ and sets $T_{ID,i}$ = $v \cdot P$ if $coin = 0$ and $T_{ID,i} = v \cdot bP$, otherwise. Finally, B adds $<$ ID, i, $T_{ID,i}$, $v, coin$ $>$ in L_1 and returns $H_1(ID, i) = T_{ID,i}$ to A_I. Indeed, when $coin$ = 1, the third value bP of the DBDH problem is put in the corresponding query $H_1(ID, i)$. On the contrast, if $coin$ = 0, the corresponding query $H_1(ID, i)$ does not include the DBDH problem. Note that the probability Pr[$coin$ = 0] will be determined later. If we place the DBDH problem on every H_1 response, then the adversary cannot issue any $Timekeyupdate$ query because the challenger cannot answer the correct time update key. In such a case, it cannot simulate the real adversary's ability.
  - *H_2-queries*: The challenger B maintains a list L_2 of pairs $< X, Y >$. Upon receiving the query along with X, B performs the following steps:
    (1) If X appears in L_2, then B responds with $H_2(X) = Y$.
    (2) If X does not appear in L_2, then B randomly chooses a string $Y \in \{0, 1\}^l$. B adds $< X, Y >$ in L_2 and returns $H_2(X) = Y$ to A_I.
  - *H_3-queries*: The challenger B maintains a list L_3 of pairs $< U, V, M, ID, i, w >$. Upon receiving the query along with (U, V, M, ID, i), B performs the following steps:
    (1) If (U, V, M, ID, i) appears in L_3, then B responds with $H_3(U, V, M, ID, i) = w$.
    (2) If (U, V, M, ID, i) does not appear in L_3, then B randomly chooses a string $w \in \{0, 1\}^l$. B adds $< U, V, M, ID, i, w >$ in L_3 and returns $H_3(X) = w$ to A_I.
- *Phase* 1. A_I is able to issue three queries and B responds as follows.
  - *Identity key extract query* (ID): To respond to such a query, the challenger B first accesses the list L_0 to obtain u. Then, B sets the identity key $D_{ID} = u \cdot P_{pub}$ which is valid since the identity key D_{ID} = $u \cdot P_{pub}$ = $u \cdot \alpha \cdot P$ = $\alpha \cdot u \cdot P$ = $\alpha \cdot S_{ID}$. B returns the identity key D_{ID} to A_I.
  - *Time key update query* (ID, i): To respond to such a query, B first accesses the list L_1 to obtain v and $coin$. If $coin$ =1, B reports failure and terminates. If $coin$ =0, B sets the time update key $P_{ID,i}$=$v \cdot C_{pub}$ which is valid since the time update key $P_{ID,i}$ = $v \cdot C_{pub}$ = $v \cdot aP$ = $a \cdot v \cdot P$ = $a \cdot T_{ID,i}$. B returns $P_{ID,i}$ to A_I.
  - *Decryption query* $(C = (U, V, W), ID, i)$: To respond to such a query, B first uses $(U, V, -, ID, i, W)$ to scan the list L_3 to obtain M. If $(U, V, -, ID, i, W)$ was not found, B returns failure and terminates which means that A can guess a right output value of H_3 hash function without using random oracles. Otherwise, B return M.
- *Challenge*. In this phase, A_I issues two messages M_0, M_1, an identity ID^* and a period i^*. If ID^* does not appear in the list L_0, B randomly chooses $u^* \in Z_q^*$ and sets S_{ID^*}= $u^* \cdot P$. B adds the tuple $<$ $ID^*, S_{ID^*}, u^* >$ in L_0. Meanwhile, B uses (ID^*, i^*) to scan the tuple $< ID^*, i^*, T_{ID^*,i^*}, v, coin >$ in the list L_1. If $coin = 0$, then B reports failure and terminates because (ID^*, i^*) is not the target identity and period. If $coin = 1$, B flips a random $\gamma \in \{0, 1\}$, receives Y^* by issuing the $H_2(\hat{e}(S_{ID^*}, \alpha \cdot cP) \cdot (v \cdot T))$ query and computes $V = M_\gamma \oplus Y^*$, where cP and T are the last two values of the DBDH problem. And B then selects a random string $w \in \{0, 1\}^l$, adds the tuple $< U = cP, V = M_\gamma \oplus Y^*, M_\gamma, ID^*, i^*, w >$ in L_3. Finally, B returns the target ciphertext C^*= $(U, V, W = w)$ to A_I.
- *Phase* 2. A_I may issue further queries as those in the $Phase$ 1. The only restriction is that A_I cannot issue the *Time key update* query with (ID^*, i^*).
- *Guess*. A_I outputs a guess γ'. The advantage ϵ of an IND-ID-CCA adversary A_I to attack the revocable IBE scheme with CRA is evaluated by the function $Adv_A = |\Pr[\gamma = \gamma'] - \frac{1}{2}|$. If the advantage ϵ of the adversary A_I is non-negligible, it means that the challenger B with non-negligible advantage can

decide if $T = \hat{e}(P, P)^{abc}$. This resolves the DBDH problem with a non-negligible probability ϵ', which will be determined later.

Next, we analyze the probability that the simulation above will not abort. In the $Phases$ 1 and 2, if $coin$ = 0, the simulation continues. For convenience, let δ denote the probability that $coin$ = 0. Since A_I makes at most q_u *Time key update* queries in the $Phases$ 1 and 2, the probability that the simulation does not abort is δ^{q_u}. In the $Challenge$ phase, if $coin$=1, the simulation continues, so the probability that the simulation does not abort is $1 - \delta$. As a result, the total probability of the simulation not aborting is $\delta^{q_u} \cdot (1 - \delta)$ in the $Phase$ 1, $Phase$ 2 and $Challenge$ phases. By a similar technique in [28], we have that the maximum value of $\delta^{q_u} \cdot (1 - \delta)$ is achieved at $\delta = 1 - 1/(q_u + 1)$ and so the probability of the simulation not aborting is at least $1/e(1 + q_u)$, where e is the base value of the natural logarithm. For handling the decryption query, if $(U, V, -, ID, i, W)$ cannot be found in the list L_3, B returns failure and terminates, which means that A_I can guess a right output value of H_3 hash function. In this case, there are q_d *decryption queries*, the probability of A_I is at most q_d/q. In summary, B can solve the DBDH problem with probability $\epsilon' \geq \frac{\epsilon}{e(1+q_u)} - \frac{q_d}{q}$. □

Lemma 2. *In the random oracle model, suppose that there is a Type II adversary A_{II} with probability ϵ who can break the proposed revocable IBE scheme with CRA in Game 2. In the meantime, let q_e and q_d denote, respectively, the numbers of identity key extract queries and decryption queries that A_{II} is allowed to issue. Then we can construct an algorithm B who solves the DBDH problem with probability*

$$\epsilon' \geq \frac{\epsilon}{e(1 + q_e)} - \frac{q_d}{q},$$

where e is the base value of the natural logarithm.

Proof. Assume that there is a Type II adversary A_{II} with probability ϵ who can break the proposed revocable IBE scheme with CRA. We construct an algorithm B to solve the DBDH problem with probability ϵ'. The algorithm B takes as input the DBDH parameters $< q, G, G_T, \hat{e} >$ and and a tuple $< P, aP, bP, cP, T >$, where P is a generator of the group G, $a, b, c \in Z_q^*$ are unknown to B and $T \in G_T$. Next, the algorithm B attempts to solve the DBDH problem by deciding if $T = \hat{e}(P, P)^{abc}$. Here, B acts as the challenger and interacts with A_{II} in Game 2 as follows.

- *System setup*. The challenger B first chooses a random master time key $\beta \in Z_q^*$ and sets $C_{pub} = \beta \cdot P \in G$. B then provides A_{II} with public parameters PP = $< q, G, G_T, \hat{e}, P, P_{pub}, C_{pub}, H_0, H_1, H_2 >$, where $P_{pub} = aP$. Moreover, H_0, H_1 and H_2 are random oracles controlled by B defined as below.
  - *H_0-queries*: The challenger B maintains a list L_0 of tuples $< ID, S_{ID}, u, coin >$. Upon receiving the query along with ID, B performs the following steps:
    (1) If ID appears in L_0, then B responds with S_{ID}.
    (2) If ID does not appear in L_0, B chooses a random value $u \in Z_q^*$. B then flips a random $coin \in \{0, 1\}$ and sets S_{ID}= $u \cdot P$ if $coin = 0$ and $S_{ID} = u \cdot bP$, otherwise. Finally, B adds $< ID, S_{ID}, u, coin >$ in L_0 and returns $H_0(ID) = S_{ID}$ to A_{II}. Indeed, when $coin$ = 1, the third value bP of the DBDH problem is put in the corresponding query $H_0(ID)$. On the contrast, if $coin$ = 0, the corresponding query $H_0(ID)$ does not include the DBDH problem. Note that the probability Pr[$coin$ = 0] will be determined later.
  - *H_1-queries*: The challenger B maintains a list L_1 of tuples $< ID, i, T_{ID,i}, v >$. Upon receiving the query along with (ID, i), B performs the following steps:
    (1) If (ID, i) appears in L_1, then B responds with $H_1(ID, i)$ = $T_{ID,i}$.
    (2) If (ID, i) does not appear in L_1, then B chooses a random value $v \in Z_q^*$ and computes $T_{ID,i} = v \cdot P$. B adds $< ID, i, T_{ID,i}, v >$ in L_1 and returns $H_1(ID, i)$ = $T_{ID,i}$ to A_{II}.
  - *H_2-queries*: The challenger B maintains a list L_2 of pairs $< X, Y >$. Upon receiving the query along with X, B performs the following steps:
    (1) If X appears in L_2, then B responds with $H_2(X)$ = Y.
    (2) If X does not appear in L_2, then B chooses a string $Y \in \{0, 1\}^l$. B adds $< X, Y >$ in L_2 and returns $H_2(X) = Y$ to A_{II}.
  - *H_3-queries*: As the *H_3-queries* in Lemma 1.
- *Phase* 1. A_{II} may issue three queries and B responds as follows.
  - *Identity key extract query* (ID): To respond to such a query, the challenger B first accesses the list L_0 to obtain u and $coin$. If $coin$ = 1, B reports failure and terminates. If $coin$ = 0, B sets the identity key D_{ID} = $u \cdot P_{pub}$ which is valid since D_{ID} = $u \cdot P_{pub}$ = $u \cdot aP$ = $a \cdot u \cdot P$ = $a \cdot S_{ID}$. B returns the identity key D_{ID} to A_{II}.
  - *Time key update query* ((ID, i)): To respond to such a query, the challenger B first accesses the list L_1 to obtain v. Then B sets the time update key $P_{ID,i}$ = $v \cdot C_{pub}$ which is valid since $P_{ID,i}$ = $v \cdot C_{pub}$ = $v \cdot \beta \cdot P$ = $\beta \cdot v \cdot P$ = $\beta T_{ID,i}$. B returns the time update key $P_{ID,i}$ to A_{II}.
  - *Decryption query* ($C = (U, V, W), ID, i$): As the *Decryption query* in Lemma 1.
- *Challenge*. In this phase, A_{II} issues two messages M_0, M_1, an identity ID^* and a period i^*. If (ID^*, i^*) does not appear in the list L_1, B randomly chooses $v^* \in Z_q^*$ and sets $T_{ID^*,i^*} = v^* \cdot P$. B adds the tuple $< ID^*, i^*, T_{ID^*,i^*}, v^* >$ in L_1 and returns $H_1(ID^*, i^*)$ = T_{ID^*,i^*} to A_{II}. Meanwhile, B uses

ID^* to scan the tuple $< ID^*, S_{ID^*}, u, coin >$ in the list L_0. If $coin = 0$, then B reports failure and terminates because ID^* is not the target identity. If $coin = 1$, B flips a random $\gamma \in \{0,1\}$, receives Y^* by issuing the $H_2((u \cdot T) \cdot \hat{e}(T_{ID^*,i^*}, \beta \cdot cP))$ query and computes $V = M_\gamma \oplus Y^*$, where cP and T are the last two values of the DBDH problem. And B then selects a random string $w \in \{0,1\}^l$, adds the tuple $< U = cP, V = M_\gamma \oplus Y^*, M_\gamma, ID^*, i^*, w >$ in L_3. Finally, B returns the target ciphertext C^*= $(U, V, W = w)$ to A_I.

- *Phase* 2. A_{II} is able to issue further queries as those in the $Phase$ 1. The only restriction is that A_{II} cannot issue the *identity key extract* query with ID^*.
- *Guess.* A_{II} outputs a guess γ'. The advantage ϵ of an IND-ID-CCA adversary A_{II} to attack the revocable IBE scheme with CRA is evaluated by the function $Adv_A = |\Pr[\gamma = \gamma'] - \frac{1}{2}|$. If the advantage ϵ of the adversary A_{II} is non-negligible, it means that the challenger B with non-negligible advantage can decide if $T = \hat{e}(P,P)^{abc}$. This resolves the DBDH problem with a non-negligible probability ϵ', which will be determined later.

Next, we analyze the probability that the simulation above will not abort. In the $Phase$ 1 and $Phase$ 2, if $coin$ = 0, the simulation continues. For convenience, let δ denote the probability that $coin$ = 0. Since A_{II} makes at most q_e *Identity key extract* queries in the $Phases$ 1 and 2, the probability that the simulation does not abort is δ^{q_e}. In the $Challenge$ phase, if $coin$ = 1, the simulation continues, so the probability that the simulation does not abort is $1 - \delta$. As a result, the total probability of the simulation not aborting is $\delta^{q_e} \cdot (1 - \delta)$ in the $Phase$ 1, $Phase$ 2 and $Challenge$ phase. As mentioned in the proof of Lemma 1, the maximum value of the probability $\delta^{q_e} \cdot (1 - \delta)$ is achieved at $\delta = 1 - 1/(q_e + 1)$ and so the probability of the simulation not aborting is at least $1/e(1 + q_e)$, where e is the base value of the natural logarithm. For handling the decryption query, if $(U, V, -, ID, i, W)$ cannot be found in the list L_3, B returns failure and terminates, which means that A_{II} can guess a right output value of H_3 hash function. In this case, there are q_d *decryption queries*, the probability of A_{II} is at most q_d/q. In summary, B can solve the DBDH problem with probability $\epsilon' \geq \frac{\epsilon}{e(1+q_e)} - \frac{q_d}{q}$. □

Theorem 3. *In the random oracle model, the proposed revocable IBE scheme with CRA is semantically secure against adaptive chosen-ciphertext attack (IND-ID-CCA) under the DBDH assumption.*

Proof. By Lemmas 1 and 2, we can conclude the theorem. □

6 Comparisons

In this section, we make comparisons between Li *et al.*'s scheme [24] and ours. Table 2 lists the notations used in evaluating the computational costs of the related pairing-based operations. By some previous implementations [29], [30], [31], we know that TG_a, T_m and T_H are negligible in comparison with the other time-consuming operations.

TABLE 2: Notations for computational costs

Notation	Operation
TG_p	A bilinear pairing $\hat{e} : G \times G \to G_T$
TG_m	A scalar multiplication in G
T_e	An exponentiation in G_T
TG_H	A map-to-point hash function
TG_a	An addition in G
T_m	A multiplication operation in G_T
T_H	A hash function
$\|C\|$	The bit length of ciphertext C

TABLE 3: Configurations of two processors

Processor	Clock speed	Configurations
Intel Core-2 Quad CPU	2.4 GHz	3 GB RAM
Q6600 computer		OS: Ubuntu 10.04
HTC Desire HD A9191	Qualcomm 1 GHz	768MB RAM
Smartphone		OS: Android 2.2

TABLE 4: Computational time for related operations on two processors

Notation	Intel Core-2 Quad CPU	HTC Desire HD A9191
	Q6600 computer	smartphone
TG_p	7.5ms	0.26s
TG_m	2.8ms	0.034s
T_e	2.1ms	0.021s
TG_H	≈2.8ms	≈0.034s

For the fairness and convenience of comparisons, we use the benchmark results implemented by Java pairing based cryptography library (JPBC) [32] to compare performance between between Li *et al.*'s scheme [24] and ours. In the benchmark results [32] , two processors on the Intel Core-2 computer and HTC Desire HD-A9191 smartphone are employed to simulate the computational costs of the cloud revocation authority (CRA) and mobile users, respectively. Table 3 lists the detailed configurations. In the meantime, a popular and valid choice for bilinear pairings would be to adopt an elliptic curve over a finite field $E(Fp)$ with a large prime p of 512 bits and a prime order q of 160 bits. The benchmark results of the related operations on the processors of the Intel Core-2 computer and HTC Desire HD-A9191 smartphone are summarized in Table 4.

In Table 5, we demonstrate the comparisons between Li *et al.*'s scheme [24] and ours in terms of computational costs, number of secret keys and bit length of ciphertext. Here, we refer to low-power computing devices (i.e., HTC Desire HD-A9191 smartphone). In contrast, both the CRA in our scheme and the KU-CSP in Li *et al.*'s scheme are regarded as powerful devices (i.e., Intel Core-2 computer).

For the time key update and the encryption procedures, two schemes possess almost the same performance. For the computational cost of the encryption, our scheme requires only TG_p, but Li *et al.*'s scheme requires $4TG_p + 4TG_m$. Note that Li *et al.*'s scheme [24] is an IND-ID-CPA-secure IBE scheme. Their scheme must use the transformation methods in [26], [27] to transform an IND-ID-CPA-secure IBE scheme

TABLE 5: Performance comparisons between Li *et al.*'s scheme and ours

	Li *et al.*'s scheme	Our scheme
Computational cost for time update key	$TG_H + 3T_e$	$TG_H + TG_m$
	9.1 (ms)	5.6 (ms)
Number of keys stored in the cloud authority	n	1
Computational cost for encryption	$TG_p + 2TG_H + TG_m + 4T_e$	$2TG_p + 2TG_H + TG_m + T_e$
	0.446 (s)	0.643 (s)
Computational cost for decryption	$4TG_p + 4TG_m$	TG_p
	1.176 (s)	0.26 (s)
Bit length of ciphertext	$\lvert G\rvert + 3\lvert G_T\rvert + l$	$\lvert G\rvert + 2l$
	512 bytes	168 bytes
	$46.4mJ$	$15.2mJ$

into an IND-ID-CCA-secure IBE scheme. In such a case, the ciphertext of their scheme have to add a hash value W in ciphertext as our proposed scheme. For the bit length of ciphertext, as mentioned earlier, a popular and valid choice for bilinear pairings would be to adopt an elliptic curve over a finite field $E(Fp)$ with a large prime p of 512 bits and a prime order q of 160 bits. In such a case, $|G| + 2l$ (168 bytes) required in our scheme is less than $|G| + 3|G_T| + l$ (512 bytes) required in Li *et al.*'s scheme, where $l = 160$ bits is the output bit length of the hash functions $H_2()$ and $H_3()$. Moreover, according to [33], transmitting 32 bytes data requires 9 bytes for the header and 8 bytes for preamble so that a packet size is 49 bytes. Meanwhile, transmitting such a packet requires about $2.9mJ$ of energy [33]. Therefore, our scheme requires $(168/32) * 2.9 = 15.2mJ$ while Li *et al.*'s scheme requires $(512/32) * 2.9 = 46.4mJ$. For scalability, the KU-CSP in Li *et al.*'s scheme must keep n various time keys for n users so that it does not possess scalability and incurs the management load. On the contrast, the CRA in our scheme holds only one master time key for all the users. When the number n of users in the system is very large, the PKG may designate multiple CRAs to share the responsibility of user revocation while each CRA holds only the same master time key. However, in Li et al.'s scheme, each KU-CSP must also keep n time keys. It is obvious that our scheme possesses not only scalability, but also better performance of computation and communication as compared to Li *et al.*'s scheme.

7 Cloud computing applications

In this section, we extend our revocable IBE scheme to discuss two extended cloud computing applications, namely, the revocable attribute-based encryption for cloud storage and the CRA-aided authentication with period-limited privileges for managing a large number of various cloud services.

7.1 Revocable attribute-based encryption

With the rapid development in wireless communication, cloud storage services [34] have become popular increasingly. Users can store their data on the cloud storage so that they may access their data anywhere at any time. Typically, the data stored on the cloud storage is encrypted for user privacy while protecting from access by other users. Indeed, due to the collaborative property of some applications, a data owner allows specific parties to decrypt the encrypted data stored on the cloud storage. In such a situation, enforcing this kind of access control by ordinary public key encryption (ex. IBE) schemes is not very convenient because it cannot provide the flexibility of users to share their data. Attribute-based encryption (ABE) [35] is regarded as one of the most suitable encryption schemes for data sharing of cloud storage. Indeed, ABE is encryption for privileges, not for users so that an ABE scheme is a very useful tool for cloud storage services since data sharing is an important feature for such services.

In 2005, Sahai and Waters [35] first introduced the concept of attribute-based encryption (ABE) which refines IBE scheme [2] by associating ciphertexts and a set of attributes. In an ABE scheme, the PKG typically sends the corresponding attribute keys for the user with several attributes. An ABE scheme allows a data owner to encrypt data under a set of attributes associated with access structures, and users who own these corresponding attribute keys are able to decrypt the encrypted data. Afterward, there are numerous ABE schemes [36], [37], [38], [39] that have been proposed. Indeed, we may combine the revocability concept of the proposed revocable IBE scheme with the existing ABE schemes to construct revocable ABE schemes. Indeed, Li *et al.* [40] and Qian *et al.* [41], respectively, proposed an ABE scheme with user/attribute revocation for various applications. Both schemes still adopt the sub-tree method in [14] to address the revocation rekeying issue so that a secure channel is used to transmit the new updated user keys and attribute keys.

For constructing such revocable ABE schemes using a public channel, we may employ the same role of the CRA to be responsible for periodically generating the attribute-time keys for users and send them to users via a public channel. The functionality of the attribute-time key is the same with that of the time update key in the proposed revocable IBE scheme. Therefore, if a data owner encrypts data under a set of attributes associated with access structures and a time period. Thus, users who own both the attribute keys and valid attribute-time keys at the time period are able to decrypt the encrypted data. If a particular attribute of a user is revoked, the CRA simply stops issuing the new corresponding attribute-time key for the user. Therefore, a revocable ABE scheme provides more flexible than an ABE scheme for managing attributes of users.

7.2 CRA-aided authentication scheme with period-limited privileges

An authentication scheme is a cryptographic mechanism to authenticate users over public networks. Before a user gains access to a server's services, the user must be authenticated and authorized by the server. Here, we extend our revocable IBE scheme to construct a cloud-revocation-authority (CRA)-aided authentication scheme with period-limited privileges for managing a large number of various cloud services [34]. When a company (or organization) constructs numerous various cloud services, how to efficiently manage the authorizations for these cloud services is an important issue since a user must authenticate herself/himself to a cloud service server before accessing the cloud services. In the system with multiple cloud services, multiple CRAs replace the role of the CRA in our proposed scheme. The master time key is replaced with multiple master privilege keys. A CRA with a master privilege key can manage the corresponding privilege to have access to some service server at various periods. A CRA is able to use its master privilege key to generate and send a period-limited privilege key to a user. A user with both the associated identity key and a period-limited privilege key is able to access the corresponding server. Indeed, a CRA may also manage single or multiple service servers. Without loss of generality, we assume that there are k independent CRAs that are responsible for managing k independent service servers, respectively.

For simplicity, we illustrate the case k = 2 by Fig. 4. The PKG randomly selects k different master privilege keys $\beta_1, \beta_2, \ldots, \beta_k$ and sends each β_j to the corresponding CRA$_j$, respectively. Also, the PKG sends the identity key D_{ID} to a legitimate user with identity ID via a secure channel. On the other hand, if this user with identity ID is granted to have access to the service server j at period i, the CRAj will use the master privilege key β_j to generate the period-limited privilege key $P_{ID,j,i}$ and send it to the user via a public channel. Consequently, the user is able to access the server j at period i by using both the identity key D_{ID} and period-limited privilege key $P_{ID,j,i}$. Note that, indeed, a CRA may manage all the privileges for all the service servers. In such a case, all the master privilege keys are sent to the designated CRA.

In the system with multiple cloud services, a user with both the identity key D_{ID} and period-limited privilege key $P_{ID,j,i}$ may run an authentication scheme, called CRA-aided authentication scheme with period-limited privileges, to authenticate herself/himself to the service server j at period i. The proposed CRA-aided authentication scheme with period-limited privileges depicted in Fig. 5, which consists of four algorithms :

- *System setup*: As in the revocable IBE scheme with CRA proposed in Section 3, a trusted PKG generates the master secret key α and computes the system public key $P_{pub} = \alpha \cdot P$. In addition, suppose that there are k independent service servers managed by k independent CRAs in the system. The PKG randomly selects k different master privilege keys $\beta_1, \beta_2, \ldots, \beta_k$ and sends each β_j to the corresponding CRA$_j$ via a secure channel, respectively. In the meantime, the PKG also computes the privilege public key $C_{pub,j} = \beta_j \cdot P$ for each CRA$_j$. The PKG selects four hash functions $H_0, H_1 : \{0,1\}^* \rightarrow G$, $H_2 : G_T \rightarrow \{0,1\}^l$, $H_3 : \{0,1\}^* \rightarrow \{0,1\}^l$, where l is fixed. Finally, the PKG publishes the public parameters $PP = < q, G, G_T, \hat{e}, P, P_{pub}, C_{pub,1}, C_{pub,2}, \ldots, C_{pub,k}, H_0, H_1, H_2, H_3 >$.
- *Identity key extract*: As in the revocable IBE scheme with CRA proposed in Section 3. Upon receiving the identity $ID \in \{0,1\}^*$ of a user, the PKG sends the identity key D_{ID} to the user via a secure channel.
- *Privilege key extract*: Suppose that a user with identity $ID \in \{0,1\}^*$ is granted to have access to the service server j at period i. The corresponding CRA$_j$ uses the master privilege key β_j to generate the period-limited privilege key $P_{ID,j,i}$=$\beta_j \cdot H_1(ID, i)$ and send it to the user via a public channel.
- *Authentication*: If a user with identity ID would like to access some service server j, the user sends an authentication request along with ID and period i to the service server j.
  - Upon receiving the authentication request, the service server j selects a challenge message $M \in \{0,1\}^l$ and a random value $r \in Z_q^*$, and computes $U = r \cdot P$ and $V = M \oplus H_2((g_1 \cdot g_2)^r)$, where $g_1 = \hat{e}(S_{ID}, P_{pub}) = \hat{e}(H_0(ID), \alpha \cdot P)$ and $g_2 = \hat{e}(T_{ID,i}, C_{pub,j}) = \hat{e}(H_1(ID,i), \beta_j \cdot P)$. Finally, the service server j sends $C = (U, V)$ to the user.
  - Upon receiving $C = (U, V)$, the user with the identity key D_{ID} and period-limited privilege key $P_{ID,j,i}$ computes $M = V \oplus H_2(\hat{e}(D_{ID} + P_{ID,j,i}, U))$. The user then sends $R = H_3(M, U, V, ID, i)$ to the service server j.
  - Upon receiving the response message R, the service server j validates whether R is equal to $H_3(M, U, V, ID, i)$ or not. If so, the service server j accepts the request, and reject, otherwise.

Indeed, authentication (identification) schemes [42], [43], [44], [45] may be implemented by signature or encryption schemes. In our authentication procedure, the service server verifies a user by asking the user to decrypt a challenge ciphertext C. Then the user responds with R, which can pass the server's verification only when the user retrieves the valid plaintext M. The proposed CRA-aided authentication scheme with period-limited privileges aims at user identification and authorization before accessing service servers. The CRA-aided authentication scheme does not concern with the construction of secure session keys for encryption. Hence, some existing session key exchange protocol [46] or SSL protocol [47] can be employed to establish a secure session key for providing communication confidentiality.

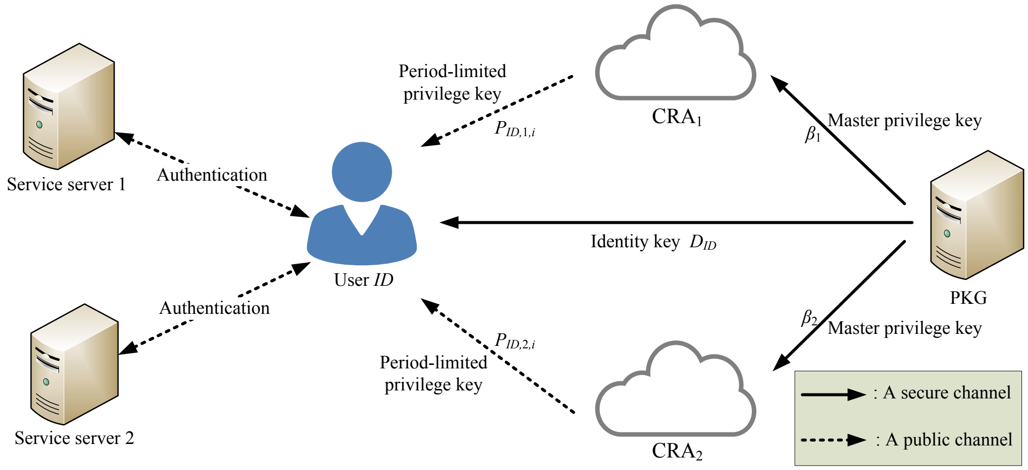


Fig. 4: Example of system model for managing multiple cloud services

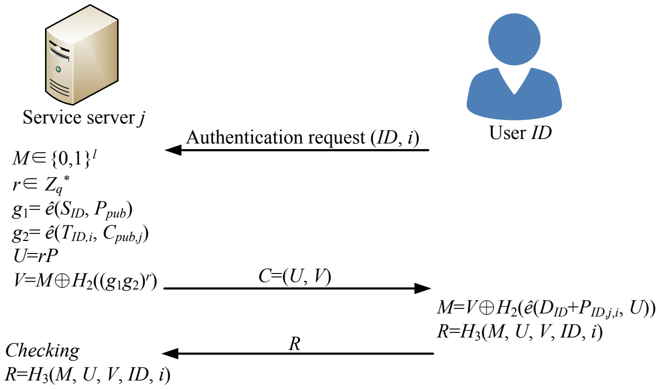

Service server j
User ID
Authentication request (ID, i)
$M\in\{0,1\}^l$
$r\in Z_q^*$
$g_1=\hat{e}(S_{ID}, P_{pub})$
$g_2=\hat{e}(T_{ID,i}, C_{pub,j})$
$U=rP$
$V=M\oplus H_2((g_1g_2)^r)$
$C=(U, V)$
$M=V\oplus H_2(\hat{e}(D_{ID}+P_{ID,j,i}, U))$
$R=H_3(M, U, V, ID, i)$
R
Checking
$R=H_3(M, U, V, ID, i)$

Fig. 5: Authentication procedure

In the following, based on the IND-ID-CCA security of the revocable IBE scheme with CRA, we prove that the proposed CRA-aided authentication scheme with period-limited privileges is secure under active attacks.

Theorem 4. *Based on the security of the revocable IBE scheme with CRA, the proposed CRA-aided authentication scheme with period-limited privileges is secure under active attacks.*

Proof Sketch. Assume that an adversary E can break the proposed CRA-aided authentication scheme with period-limited privileges. We will use E to construct an algorithm F that wins the IND-ID-CPA games (Games 1 and 2) of the revocable IBE scheme with CRA, in which the algorithm F plays the roles of adversaries A_I and A_{II}. In the $Challenge$ phase of Games 1 and 2, the adversary F selects and sends a plaintext pair (M_0, M_1) to the challenge B. The challenge B flips a random coin $\gamma \in \{0, 1\}$, sets the ciphertext C^* = $E(ID^*, i^*, M_\gamma)$ and returns C^* to the adversary F. Upon receiving C^*, the adversary F plays the role of service server to obtain the response R^* from the adversary E in the proposed CRA-aided authentication scheme. The adversary F checks R^* = $H_3(M_0, U^*, V^*, ID^*, i^*)$ or R^* = $H_3(M_1, U^*, V^*, ID^*, i^*)$. Hence, in the $Guess$ phase of Games 1 and 2, F always outputs a correct bit $\gamma' = \gamma$. We say that the adversary F wins the IND-ID-CCA games (Games 1 and 2). This contradicts Theorem 3. □

8 Conclusions

In this article, we proposed a new revocable IBE scheme with a cloud revocation authority (CRA), in which the revocation procedure is performed by the CRA to alleviate the load of the PKG. This outsourcing computation technique with other authorities has been employed in Li *et al.*'s revocable IBE scheme with KU-CSP. However, their scheme requires higher computational and communicational costs than previously proposed IBE schemes. For the time key update procedure, the KU-CSP in Li *et al.*'s scheme must keep a secret value for each user so that it is lack of scalability. In our revocable IBE scheme with CRA, the CRA holds only a master time key to perform the time key update procedures for all the users without affecting security. As compared with Li *et al.*'s scheme, the performances of computation and communication are significantly improved. By experimental results and performance analysis, our scheme is well suited for mobile devices. For security analysis, we have demonstrated that our scheme is semantically secure against adaptive-ID attacks under the decisional bilinear Diffie-Hellman assumption. Finally, based on the proposed revocable IBE scheme with CRA, we constructed a CRA-aided authentication scheme with period-limited privileges for managing a large number of various cloud services.

Acknowledgments

The authors would like to appreciate anonymous referees for their valuable comments and constructive suggestions. This research was partially supported by Ministry of Science and Technology, Taiwan, under contract no. MOST103-2221-E-022-MY2.

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," Proc. Crypto'84, LNCS, vol. 196, pp. 47-53, 1984.
[2] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," Proc. Crypto'01, LNCS, vol. 2139, pp. 213-229, 2001.

[3] R. Housley, W. Polk, W. Ford, and D. Solo, “Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile,” IETF, RFC 3280, 2002.

[4] W. Aiello, S. Lodha, and R. Ostrovsky, “Fast digital identity revocation,” Proc. Crypto’98, LNCS, vol. 1462, pp. 137-152, 1998.

[5] M. Naor and K. Nissim, “Certificate revocation and certificate update,” IEEE Journal on Selected Areas in Communications, vol. 18 , no. 4, pp. 561 - 570, 2000.

[6] S. Micali, “Novomodo: Scalable certificate validation and simplified PKI management,” Proc. 1st Annual PKI Research Workshop, pp. 15-25, 2002.

[7] F. F. Elwailly, C. Gentry, and Z. Ramzan, “QuasiModo: Efficient certificate validation and revocation,” Proc. PKC’04, LNCS, vol. 2947, pp. 375-388, 2004.

[8] V. Goyal, “Certificate revocation using fine grained certificate space partitioning,” Proc. Financial Cryptography, LNCS, vol. 4886, pp. 247-259, 2007.

[9] D. Boneh, X. Ding, G. Tsudik, and C.-M. Wong, “A Method for fast revocation of public key certificates and security capabilities,” Proc. 10th USENIX Security Symp., pp. 297-310. 2001.

[10] X. Ding and G. Tsudik, “Simple identity-based cryptography with mediated RSA,” Proc. CT-RSA’03, LNCS, vol. 2612, pp. 193-210, 2003.

[11] B. Libert and J. J. Quisquater, “Efficient revocation and threshold pairing based cryptosystems,” Proc. PODC2003, pp. 163-171, 2003.

[12] J. Baek and Y. Zheng, “Identity-based threshold decryption,” Proc. PKC’04, LNCS, vol. 2947, pp. 262-276, 2004.

[13] H.-S. Ju, D.-Y. Kim, D.-H. Lee, H. Park, and K. Chun, “Modified ID-based threshold decryption and its application to mediated ID-based encryption,” Proc. APWeb2006, LNCS, vol. 3841, pp. 720-725, 2006.

[14] A. Boldyreva, V. Goyal, and V. Kumar, “Identity-based encryption with efficient revocation,” Proc. CCS’08, pp. 417-426, 2008.

[15] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” Proc. Eurocrypt’05, LNCS, vol. 3494, pp. 557-557, 2005.

[16] B. Libert and D. Vergnaud, “Adaptive-ID secure revocable identity-based encryption,” Proc. CT-RSA’09, LNCS, vol. 5473, pp. 1-15, 2009.

[17] J.-H. Seo and K. Emura, “Revocable identity-based encryption revisited: security model and construction,” Proc. PKC’13, LNCS, vol. 7778, pp. 216-234, 2013.

[18] S. Park, K. Lee, and D.H. Lee, “New constructions of revocable identity-based encryption from multilinear maps,” IEEE Transactions on Information Forensics and Security, vol.10 , no. 8, pp. 1564 - 1577, 2015.

[19] C. Wang, Y. Li, X. Xia, and K. Zheng, “An efficient and provable secure revocable identity-based encryption scheme,” PLoS ONE, vol. 9, no. 9, article: e106925, 2014.

[20] A. Lewko A and B. Waters, “New techniques for dual system encryption and fully secure hibe with short ciphertexts,” Proc. TCC’10, LNCS, vol. 5978, pp. 455-479, 2010.

[21] J.-H. Seo and K. Emura, “Efficient delegation of key generation and revocation functionalities in identity-based encryption,” Proc. CT-RSA’13, LNCS, vol. 7779, pp. 343-358, 2013.

[22] J.-H. Seo and K. Emura, “Revocable hierarchical identity-based encryption: history-free update, security against insiders, and short Ciphertexts,” Proc. CT-RSA’15, LNCS, vol. 9048, pp. 106-123, 2015.

[23] Y.-M. Tseng. and T.-T. Tsai, “Efficient revocable ID-based encryption with a public channel,” Computer Journal, vol.55, no.4, pp. 475-486, 2012.

[24] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, “Identity-based encryption with outsourced revocation in cloud computing,” IEEE Trans. on Computers, vol. 64, no. 2, pp. 425-437, 2015.

[25] S. Galbraith, K. Paterson, and N. P. Smart, “Pairings for cryptographers,” Discrete Applied Mathematics, vol. 156, no. 16, pp. 3113-3121, 2008.

[26] E. Fujisaki and T. Okamoto, “How to enhance the security of public-key encryption at minimum Cost,” Proc. PKC’99, LNCS, vol. 1560, pp. 53-68, 1999.

[27] T. Kitagawa, P. Yang, G. Hanaoka, R. Zhang, K. Matsuura, and H. Imai, “Generic transforms to acquire CCA-security for identity based encryption: The Cases of FOPKC and REACT,” Proc. ACISP’06, LNCS, vol. 4058, pp. 348-359, 2006.

[28] J. S. Coron, “On the exact security of full domain hash,” Proc. Crypto’00, LNCS, vol. 1880, pp. 229-235, 2000.

[29] M. Scott, “Computing the Tate pairing,” Proc. CT-RSA’05, LNCS, vol. 3376, pp. 293-304, 2005.

[30] M. Scott, N. Costigan, and W. Abdulwahab, “Implementing cryptographic pairings on smartcards,” Proc. CHES’06, LNCS, vol. 4249, pp. 134-147, 2006.

[31] T.-Y. Wu and Y.-M. Tseng, “An efficient user authentication and key exchange protocol for mobile client-server environment,” Computer Networks, vol. 54, no. 9, pp. 1520-1530, 2010.

[32] B. Lynn (2015), Java Pairing Based Cryptography Library (JPBC) [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/benchmark.html

[33] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, ”Energy analysis of public-key cryptography for wireless sensor networks,” Proc. 3rd IEEE International Conf. Pervasive Computing Commun, pp. 324-328, 2005.

[34] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, ”A view of cloud computing,” Commun. ACM, vol. 53, no. 4, pp. 50-58, 2010.

[35] A. Sahai and B. Waters, ”Fuzzy identity-based encryption,” Proc. Eurocrypt’05, LNCS, vol. 3493, pp. 457-473, 2005.

[36] V. Goyal, O. Pandey, A. Sahai, and B. Waters, ”Attribute-based encryption for fine-grained access control of encrypted data,” Proc. ACM CCS, pp. 89-98, 2006.

[37] A. Sahai, H. Seyalioglu, and B. Waters, ”Dynamic credentials and ciphertext delegation for attribute-based encryption,” Proc. Crypto’12, LNCS, vol. 7417 , pp. 199-217, 2012.

[38] S. Hohenberger and B. Waters, ”Attribute-based encryption with fast decryption,” Proc. PKC’13, LNCS, vol. 7778, pp. 162-179, 2013.

[39] P.-W. Chi and C.-L. Lei, ”Audit-free cloud Storage via deniable attribute-based encryption,” IEEE Transactions on Cloud Computing, article in press (DOI: 10.1109/TCC.2015.2424882), 2015.

[40] J. Li, Y. Shi, and Y. Zhang, ”Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage,” International Journal of Communication Systems, article in press (DOI: 10.1002/dac.2942), 2015.

[41] H. Qian, J. Li, Y. Zhang, and J. Han, ”Privacy preserving personal health record using multi-authority attribute-based encryption with revocation,” International Journal of Information Security, vol. 14, no. 6, pp. 487-497, 2015.

[42] A. Fiat and A. Shamir, “How to prove yourself: practical solutions to identification and signature Problems,” Proc. Crypto’ 86, LNCS, vol. 263, pp. 186-194, 1987.

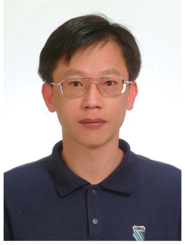
[43] K. Kurosawa and S. Heng, “From digital signature to ID-based identification/signature,” Proc. PKC’04, LNCS, vol. 2947, pp 248-261, 2004.

[44] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, “Strong authentication for RFID systems using the AES algorithm,” Proc. CHES’04, LNCS, vol. 3156, pp. 357-370, 2004.

[45] Y.-M. Tseng, T.-Y. Wu, and J.-D. Wu, “A pairing-based user authentication scheme for wireless clients with smart cards,” Informatica, vol. 19, no. 2, pp. 285-302, 2008.

[46] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, “Internet key exchange protocol version 2 (IKEv2) ,” IETF, RFC 7296, 2014.

[47] A. Freier, P. Karlton, and P. Kocher, “The secure sockets layer (SSL) protocol version 3.0,” IETF, RFC 6101, 2011.



Yuh-Min Tseng received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 1988; the M.S. degree from National Taiwan University, Taipei, Taiwan, in 1990 and the Ph.D. degree from National Chung Hsing University, Taichung, Taiwan, in 1999. He is currently a Professor with the Department of Mathematics, National Changhua University of Education, Changhua, Taiwan. His research interests include cryptography, network security, computer network, and mobile communications. Prof. Tseng is a member of the IEEE Computer Society, IEEE Communications Society, and the Chinese Cryptology and Information Security Association (CCISA). In 2006, he was the recipient of the Wilkes Award from the British Computer Society. He has published over 100 scientific journal and conference papers on various research areas of cryptography, security and computer network. He serves as an Editor of several international journals.

Tung-Tso Tsai received the B.S. degree from the Department of Applied Mathematics, Chinese Culture University, Taiwan, in 2006. He received the M.S. degree from the Department of Applied Mathematics, National Hsinchu University of Education, Taiwan, in 2009. He received the Ph.D. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2014. He is currently a senior research engineer in Hon-Hai Technology Group. His research interests include applied cryptography, pairing-based cryptography and network security.

Sen-Shan Huang is currently a Professor in the Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include number theory, cryptography, and network security. He received his Ph.D. from the University of Illinois at Urbana-Champaign under the supervision of Professor Bruce C. Berndt. Berndt.

Chung-Peng Huang received the B.S. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2013. He received the M.S. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2015. His research interests include applied cryptography and network security.