ISSN 1392 – 124X, ISSN 2335 – 884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2013, Vol. 42, No. 1

Efficient Revocable Multi-Receiver ID-Based Encryption

Tung-Tso Tsai[1], Yuh-Min Tseng[1,*], Tsu-Yang Wu[2]

[1] *Department of Mathematics, National Changhua University of Education, Jin-De Campus, Chang-Hua City 500, Taiwan*

[2] *School of Computer Science and Technology, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, P.R. China*
e-mail: ymtseng@cc.ncue.edu.tw

crossref http://dx.doi.org/10.5755/j01.itc.42.2.2244

Abstract. Quite recently, Tseng and Tsai proposed a revocable identity (ID)-based encryption (RIBE) with a public channel, in which the private key generator (PKG) can efficiently revoke misbehaving/compromised users by using a public channel. Considering the problem where a sender would like to encrypt an identical message for n receivers, the sender must re-encrypt the message n times using Tseng and Tsai's RIBE scheme. In such a case, n expensive pairing operations are required for the re-encrypting procedure. In this paper, for reducing the pairing operations, we extend Tseng and Tsai's RIBE to propose an efficient revocable multi-receiver ID-based encryption (RMIBE) scheme. Our scheme only needs one pairing operation to encrypt an identical message for n receivers while remaining the merit of user revocability in Tseng and Tsai's RIBE scheme. We demonstrate that the RMIBE scheme is semantically secure against adaptive chosen ciphertext attacks (CCA) in the random oracle model.

Keywords: revocation, multi-receiver, ID-based encryption, bilinear pairing, random oracle model.

1. Introduction

The concept of identity (ID)-based encryption was first presented by Shamir [1]. A user's identity (*e.g.* name, e-mail address or social security number) may be viewed as the user's public key. This approach can eliminate the need of certificates that make publicly available the mapping between identities and public keys. However, Shamir's construction suffers from implementing and security problems. Until 2001, Boneh and Franklin [2] defined the formal security model of ID-based encryption (IBE) and proposed the first practical IBE scheme from the Weil pairing defined on supersingular elliptic curves or abelian varieties. Subsequently, the study of ID-based cryptography has received a great attention from researchers and a large number of ID-based cryptographic schemes and protocols have been published [3-13].

Any public key system must provide a revocation mechanism to remove misbehaving/compromised users from the systems. Since the ID-based public key systems eliminate the need of certificate management, the revoking method of certificate revocation list (CRL) [14] used in certificated-based public key systems will not be the good solution to the ID-based system. For the revocation problem, Boneh and Franklin [2] also suggested a revocation mechanism, in which all non-revoked users must obtain new private keys for each period. Thus, a secure channel must be established between the private key generator (PKG) and each non-revoked user to transmit the periodic private keys. In such a case, the PKG and each non-revoked user must encrypt and decrypt the periodic private keys, respectively. In addition, the total size of key update grows linearly with the number of non-revoked users.

For improving the key update load in Boneh and Franklin's IBE scheme, Boldyreva *et al.* [15] proposed a revocable ID-based encryption (RIBE) and its associated revocation solution, in which they used a binary tree structure to reduce the key update size to logarithmic in the number of non-revoked users. They proved that their RIBE is secure under an adapted version of the selective-ID model [16], in which before the system begins to be operated, the adversary has to decide which identity it would like to attack. For enhancing the security, Libert and Vergnaud [17] improved Boldyreva *et al.*'s RIBE [15] to present an

* Corresponding author

adaptive-ID secure RIBE scheme. However, both RIBE schemes still require a s ecure channel to transmit user's periodic private keys. Additionally, each user holds 3log n private keys and the PKG must maintain a binary tree of n leaf nodes, where n denotes the number of all users.

Recently, Tseng and Tsai [18] proposed an efficient revocable ID-based encryption (RIBE) scheme with a public channel. They proved that the RIBE scheme provides adaptive chosen ciphertext (CCA) security. In their scheme, the requirement of secure channel is released and the private key size kept by each user is constant. The computational costs for encryption and decryption procedures are also improved as compared to the RIBE schemes in [15, 17]. However, they did not address the problem where a sender would like to encrypt an identical message for n receivers. Certainly, the sender may re-encrypt the identical message n times using Tseng and Tsai's RIBE scheme. As a result, n expensive pairing operations are required for the re-encrypting procedure.

Considering the situation where any user can send a message to multiple identities, in this paper, we extend Tseng and Tsai's RIBE [18] scheme to propose an efficient revocable multi-receiver ID-based encryption scheme while remaining their merits of revoking misbehaving/compromised users via a public channel. We first present the framework of revocable multi-receiver ID-based encryption (RMIBE) with a public channel. Then, we define the security notions of RMIBE that formalize possible threats and attacks. Following the framework of RMIBE, a concrete construction is proposed, in which a sender only needs one pairing operation to encrypt a message for n receivers. As a result, the performance is greatly improved as compared to the construction of re-encrypting the identical message using Tseng and Tsai's RIBE scheme. For security analysis, we prove that the proposed RMIBE scheme provides adaptive chosen ciphertext (CCA) security under the gap-bilinear Diffie-Hellman assumption [20].

The remainder of the paper is organized as follows. Preliminaries are given in Section 2. In Section 3, we formally present the definitions and security notions of revocable multi-receiver ID-based encryption (RMIBE) with a public channel. The concrete RMIBE scheme is proposed in Section 4. We analyze the security of the proposed RMIBE scheme in Section 5. Section 6 demonstrates performance analysis and comparisons. Conclusions are given in Section 7.

2. Preliminaries

In this section, we briefly introduce the concept of bilinear pairings and the related mathematical assumptions. Bilinear pairings such as Weil, Tate and Ate pairings defined on elliptic curves have been used to establish efficient ID-based encryption [2, 21, 22].

2.1. Bilinear Pairings

Let G_1 and G_2 be additive and multiplicative cyclic groups of large prime order q, respectively. In particular, G_1 is a subgroup of the group of points on an elliptic curve over a finite field and G_2 is a subgroup of the multiplicative group over a finite field. Let P be a g enerator of G_1. An admissible bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ must satisfy the following properties:

(1) Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.

(2) Non-degenerate: There exist $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$.

(3) Computability: For $P, Q \in G_1$, there exists an efficient algorithm to compute $\hat{e}(P, Q)$.

We can refer to [2, 6, 19] for full descriptions of groups, maps and other parameters. The relationship between the security levels and speed of pairing computations are referred to [10, 23].

2.2. Related Mathematical Assumptions

Here, we present three mathematical problems and define two security assumptions for bilinear pairings on which our schemes are based.

- Bilinear Diffie-Hellman (BDH) problem: Given P, aP, bP, c$P \in G_1$ for unknown a, b, $c \in Z_q^*$, compute $\hat{e}(P, P)^{abc} \in G_2$.
- Bilinear Decision Diffie-Hellman (BDDH) problem: Given P, aP, bP, $cP \in G_1$ for some a, b, $c \in Z_q^*$ and $k \in G_2$, decide whether $k = \hat{e}(P, P)^{abc}$.
- Gap-Bilinear Diffie-Hellman (Gap-BDH) problem: Given P, aP, bP, $cP \in G_1$ for some a, b, $c \in Z_q^*$, compute a Bilinear Diffie-Hellman pairing $\hat{e}(P, P)^{abc}$ with the help of the Bilinear Decision Diffie-Hellman oracle.

Definition 1 (BDDH assumption) [19]**.** Given P, aP, bP, $cP \in G_1$ for some a, b, $c \in Z_q^*$ and $k \in G_2$, there exists no probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ with non-negligible probability who can decide whether $k = \hat{e}(P, P)^{abc}$. The successful probability (advantage) of the adversary $\mathcal{A}$ is presented as

$$\mathrm{Adv}_{\mathcal{A}} = \Pr[\mathcal{A}(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] - \Pr[\mathcal{A}(P, aP, bP, cP, k) = 1],$$

where $k \in G_2$ is chosen uniformly at random and the probability is over the random choice consumed by the adversary $\mathcal{A}$.

Definition 2 (Gap-BDH assumption) [20]**.** Given P, aP, bP, $cP \in G_1$ for some a, b, $c \in Z_q^*$, there exists no probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ with non-negligible probability that can compute the Bilinear Diffie-Hellman pairing $\hat{e}(P, P)^{abc}$ with the help of the Decision Bilinear Diffie-Hellman (DBDH) oracle. The successful probability (advantage) of the adversary $\mathcal{A}$ is presented as

$$\mathrm{Adv}_{\mathrm{A}} = \Pr[\mathcal{A}(P, aP, bP, cP) = \hat{e}(P, P)^{abc}],$$

where the probability is over the random choice consumed by the adversary $\mathcal{A}$.

2.3. Notations

We define the following notations that are used throughout this paper:

- $\hat{e}$: an admissible bilinear map, $\hat{e}: G_1 \times G_1 \rightarrow G_2$.
- P: a generator of the group G_1.
- s: the system secret key.
- P_{pub}: the system public key $P_{pub} = s \cdot P$.
- ID: the identity of a user.
- D_{ID}: the user's initial secret key.
- i: a time period i, where $1 \le i \le z$ and z denotes the total number of time periods.
- $T_{ID,i}$: a user's time update key for time period i.
- $D_{ID,i}$: a user's entire decryption key for time period i, where $D_{ID,i} = D_{ID} + T_{ID,i}$.
- $H_1()$: a map-to-point function, $H_1: \{0, 1\}^* \rightarrow G_1$.
- $H_2()$: a map-to-point function, $H_2: \{0, 1\}^* \rightarrow G_1$.
- $H_3()$:a hash function, $H_3: G_2 \rightarrow \{0, 1\}^x$, where x is a fixed length.
- $H_4()$:a hash function, $H_4: G_1 \times G_1 \times \ldots \times G_1 \times G_2 \times \{0, 1\}^x \rightarrow \{0, 1\}^y$, where y is a fixed length.

3. Framework and security notions of RMIBE

In [18], Tseng and Tsai presented the framework and security notions of revocable ID-based encryption (RIBE) with a public channel. Under their framework of RIBE, a user's decryption key is divided into two components including a fixed initial secret key and a changed time update key along with time period. We extend their concept to define a new framework of revocable multi-receiver ID-based encryption (RMIBE) with a public channel.

We first describe it informally. In the system, there are two roles: a trusted private key generator (PKG) and users. Without loss of generality, the whole lifetime of the system is divided into distinct time periods 1, 2, …, z. The PKG keeps a system secret key and announces the public parameters. For a given user's identity ID, the PKG computes his/her associated initial secret key and sends it to the user via a secure channel. At the beginning of each time period, the PKG uses the system secret key to generate a time update key for each non-revoked user, called the key update process. The PKG may send them to users by using a public channel (*e.g.* E-mail). For RMIBE, it is worth noting that any sender without concerning with the key update process can encrypt a message for multiple receivers during time period i. Upon receiving the ciphertext C, one selected receiver with the valid decryption key can recover the message.

3.1. Framework

In this subsection, we formally define the framework of revocable multi-receiver ID-based encryption with a public channel.

Definition 3. A revocable multi-receiver ID-based encryption (RMIBE) with a public channel has 5-tuple of polynomial time algorithms ($\mathcal{G}$, $\mathcal{IKE}$, $\mathcal{TKU}$, $\mathcal{E}$, $\mathcal{D}$) as follows:

- The *system setup algorithm* $\mathcal{G}$: The probabilistic algorithm takes as input a security parameter l and the total number z of all time periods. It returns a system secret key s and the public parameters $Parms$. The public parameters $Parms$ are made public and implicitly inputted to all the following algorithms.
- The *initial key extract algorithm* $\mathcal{IKE}$: This deterministic algorithm takes as input the system secret key s and a user's identity $ID \in \{0, 1\}^*$ and returns the user's initial secret key D_{ID}.
- The *time key update algorithm* $\mathcal{TKU}$: This deterministic algorithm takes as input the system secret key s, a user's identity $ID \in \{0, 1\}^*$ and a time period i, then returns the user's time update key $T_{ID,i}$.
- The *encryption algorithm* $\mathcal{E}$: One sender takes as input a time period i, the multiple identities ID_1, ID_2,…, ID_n, and a message m. It then generates a ciphertext C.
- The *decryption algorithm* $\mathcal{D}$: One receiver takes as input a ciphertext C and the user's entire decryption key $D_{ID,i}$. It returns a plaintext m. Note that the user's entire decryption key $D_{ID,i}$ is obtained by $D_{ID,i} = D_{ID} + T_{ID,i}$, where D_{ID} and $T_{ID,i}$ are generated by the *initial key extract algorithm* and the *time key update algorithm*, respectively.

3.2. Security Notions

For ID-based encryption, it should be semantically secure against adaptive chosen ciphertext attacks (CCA) [2]. In 2005, Baek *et al.* [20] defined the security model for multi-receiver ID-based encryption, called selective-ID version, which is a weaker security proposed by Canetti *et al*. [16] than adaptive-ID version. The selective-ID model means that before the system begins to be operated, the adversary has to decide which identities it would like to attack. Recently, Tseng and Tsai [18] defined the security model of RIBE. We modify the above definitions to say that a revocable multi-receiver ID-based encryption (RMIBE) is semantically secure against selective multi-ID, adaptive chosen ciphertext attacks (IND-sRMID-CCA) as follows.

Definition 4 (IND-sRMID-CCA). We say that a RMIBE scheme is semantically secure against selective multi-ID, adaptive chosen ciphertext attacks (IND-sRMID-CCA) if no probabilistic polynomial-time adversary $\mathcal{A}$ has a non-negligible advantage in the following IND-sRMID-CCA game played with a challenger $\mathcal{B}$.

- *Phase* 1. $\mathcal{A}$ outputs target multiple identities denoted by (ID_1^*, ID_2^*,…, ID_n^*) and a target period time denoted by i^*.
- *Setup*. The challenger $\mathcal{B}$ runs the *system setup algorithm* $\mathcal{G}$ of RMIBE to produce a system secret key s and the public parameters *Parms*. Then the challenger $\mathcal{B}$ gives *Parms* to $\mathcal{A}$ and keeps the system secret key s to itself.
- *Phase* 2. The adversary $\mathcal{A}$ may make a number of different queries to the challenger $\mathcal{B}$ as follows:
  - *Initial key extract query* (*ID*). Upon receiving this query with *ID*, the challenger $\mathcal{B}$ runs the *initial key extract algorithm* $\mathcal{IKE}$ to return the user's initial secret key D_{ID} to $\mathcal{A}$.
  - *Time key update query* (*ID*, *i*). Upon receiving this query with (*ID*, *i*), the challenger $\mathcal{B}$ runs the *time key update algorithm* $\mathcal{TKU}$ to return the user's time update key $T_{ID,i}$ to $\mathcal{A}$.
  - *Decryption query* (*ID*, *i*, *C*). Upon receiving the query, the challenger $\mathcal{B}$ accesses the entire decryption key $D_{ID,i}$. The entire decryption key $D_{ID,i}$ is implicitly obtained by issuing the *initial key extract query* (*ID*) and the *time key update query* (*ID*, *i*). The challenger $\mathcal{B}$ runs the *decryption algorithm* $\mathcal{D}$ to decrypt the ciphertext *C*. Then it returns $\mathcal{D}(D_{ID,i}, C)$ to $\mathcal{A}$. A restriction here is that (ID, i, C) $\neq$ (ID_j^*, i^*, C), for j = 1, 2,…, n.
- *Challenge*. The adversary $\mathcal{A}$ gives a target plaintext pair (m_0^*, m_1^*) to $\mathcal{B}$. The challenger $\mathcal{B}$ chooses a random $\beta \in \{0, 1\}$ and computes C^* by running the *encryption algorithm* $\mathcal{E}$(*Parms*, (ID_1^*, ID_2^*,…, ID_n^*), i^*, m_β^*). Then $\mathcal{B}$ sends C^* to $\mathcal{A}$.
- *Phase* 3. The adversary $\mathcal{A}$ may issue more queries as in *Phase* 2. A restriction is that (ID, i, C) $\neq$ (ID_j^*, i^*, C^*) for j = 1, 2,…, n.
- *Guess*. The adversary $\mathcal{A}$ outputs $\beta' \in \{0, 1\}$ and wins this game if $\beta' = \beta$.

By the above IND-sRMID-CCA game, we refer to such an adversary $\mathcal{A}$ as a polynomial-time adversary. We define the adversary $\mathcal{A}$'s advantage in attacking the RMIBE scheme as $\mathrm{Adv}_A(l) = |\Pr[\beta = \beta'] - 1/2|$.

4. Concrete RMIBE scheme

Following the framework of RMIBE defined in Section 3, here we construct the RMIBE scheme that offers the IND-sRMID-CCA security. In order to enhance the security of the proposed RMIBE scheme, we employ the technique used in the REACT scheme proposed by Okamoto and Pointcheval [24] to construct a RMIBE scheme. The proposed RMIBE scheme consists of five algorithms that include the *system setup*, the *initial key extract*, the *time key update*, the *encryption* and the *decryption*. We describe them as follows:

- *System setup*: Given a security parameter l, a trusted private key generation (PKG) generates two groups G_1, G_2 of prime order $q > 2^l$, an admissible bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ and a generator P of G_1. The PKG performs the following tasks.
  (1) Randomly choose a system secret key $s \in Z_q^*$ and compute $P_{pub} = s \cdot P \in G_1$ as the system public key.
  (2) Choose a random $Q \in G_1^*$ and pick four hash functions $H_1: \{0, 1\}^* \rightarrow G_1$, $H_2: \{0, 1\}^* \rightarrow G_1$, $H_3: G_2 \rightarrow \{0, 1\}^x$ and H_4: $G_1 \times G_1 \times \ldots \times G_1 \times G_2 \times \{0, 1\}^x \rightarrow \{0, 1\}^y$, where x and y are fixed lengths.
  Then the public parameters and functions are presented as $Parms = \{G_1, G_2, \hat{e}, P, Q, P_{pub}, H_1, H_2, H_3, H_4\}$.
- The *initial key extract*: For a given user's identity $ID \in \{0, 1\}^*$, the PKG performs the following tasks.
  (1) Compute $Q_{ID} = H_1(ID)$ and the initial secret key $D_{ID} = s \cdot Q_{ID} \in G_1$.
  (2) Transmit D_{ID} to the user via a secure channel.
- The *time key update*: For a given time period i and a non-revoked user's $ID \in \{0,1\}^*$, the PKG performs the following tasks.
  (1) Compute $R_{ID,i} = H_2(ID, i)$ and the time update key $T_{ID,i} = s \cdot R_{ID,i} \in G_1$.
  (2) Send $T_{ID,i}$ to the user via a public channel.
  Thus, the non-revoked user can compute his/her entire decryption key $D_{ID,i} = D_{ID} + T_{ID,i}$ for time period i.
- The *encryption*: In time period i, given a message m and multiple receivers with identities ID_j for j =1, 2, 3, …, n, a sender performs the following tasks.
  (1) Compute $Q_{IDj,i} = Q_{IDj} + R_{IDj,i} = H_1(ID_j) + H_2(ID_j, i)$, for j =1, 2, 3,…, n.
  (2) Choose random $r \in Z_q^*$, and then compute $U = r \cdot P$ and $V_j = r \cdot (Q_{ID_j,i} + Q)$, for j =1, 2, 3,…, n.
  (3) Randomly choose $R \in G_2$, and then compute $W_1 = \hat{e}(Q, P_{pub})^r \cdot R$ and $W_2 = m \oplus H_3(R)$.
  (4) Compute $\sigma = H_4(R, m, U, V_1, V_2, \ldots, V_n, W_1, W_2, L)$, where L contains information about how "V_j" is associated with each receiver.
  Then the ciphertext for the message m is $C = (U, V_1, V_2, \ldots, V_n, W_1, W_2, L, \sigma)$.
- The *decryption*: Given a ciphertext $C = (U, V_1, V_2, \ldots, V_n, W_1, W_2, L, \sigma)$, the receiver ID_j uses L to find the appropriate V_j. Then the receiver uses the associated V_j to perform the following tasks.
  (1) Compute $R' = \dfrac{\hat{e}(U, D_{IDj,i})}{\hat{e}(P_{pub}, V_j)} W_1$ and $m' = W_2 \oplus H_3(R')$.
  (2) Compute $\sigma' = H_4(R', m', U, V_1, V_2, \ldots, V_n, W_1, W_2, L)$.

If $\sigma' = \sigma$, the receiver returns m as a plaintext and "*Reject*" otherwise.

5. Security analysis

As mentioned in Section 3, the adversary is allowed to obtain either the initial secret key or the time update key. Since the user's entire decryption key consists of the initial secret key and the time update key, the adversary who gets one of them is still unable to compute the user's entire decryption key. For simplicity of security proof, we consider two cases: an inside adversary (or a revoked user) and an outside adversary. If the PKG stops to issue the new time update key for a user, the user is unable to obtain the time update key in the present time period. Since the user still owns the initial secret key, this user may be viewed as an insider adversary (or a revoked user). On the other hand, any user is able to obtain the time update key, since the time update key is transmitted via a public channel. This kind of attacker may be viewed as an outsider adversary. An insider adversary and an outside adversary are allowed to issue all queries in the IND-sRMID-CCA game except for the *time key update query* on (ID^*, i^*) and the *initial key extract query* on ID^*, respectively. In the following, we give formal security analysis of the proposed RIBE scheme in the random model [25, 26].

Here, we demonstrate that the proposed RMIBE scheme is semantically secure against selective multi-ID, adaptive chosen ciphertext attacks (IND-sRMID-CCA) for the outsider and insider adversary. We adopt the same proving technique as in Baek *et al.*'s MIBE scheme [20]. They first constructed a normal public key encryption, called the Bilinear ElGamal scheme. The concrete Bilinear ElGamal scheme is described as follows.

- *KeyGen*: Choose two groups G_1, G_2 of prime order q, an admissible bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ and a generator P of G_1. Choose $s \in Z_q^*$ uniformly at random and compute $P_{pub} = s \cdot P \in G_1$. Choose a random $Q \in G_1^*$. The public key is presented as $PK=\{G_1, G_2, \hat{e}, P, Q, P_{pub}\}$ and the private key are presented as $SK=\{G_1, G_2, \hat{e}, P, P_{pub}, s\}$.
- *Encrypt*: Given a message $m \in G_2$ and the public key PK, choose a random $r \in Z_q^*$ and compute $C = (U, W) = (rP, \hat{e}(Q, P_{pub})^r \cdot m)$. Return this ciphertext C.
- *Decrypt*: Given a ciphertext C and the private key SK, compute $m = \dfrac{W}{\hat{e}(U,Q)^s}$ and return m as a plaintext.

Baek *et al*. [20] proved that the above Bilinear ElGamal scheme is "One-Way-ness under plaintext checking attack" (OW-PCA) secure assuming that the Gap-BDH problem is intractable, in which the OW-PCA security was defined by Okamoto and Pointcheval [24]. We present an informal description about the OW-PCA security. Suppose that there exists a *Plaintext Checking* (*PC*) *oracle*, which, given a ciphertext-plaintext message pair (C, M), outputs 1 if C encrypts M and 0 otherwise. We say that a public key encryption scheme is (t', q_o, ε')-OW-PCA secure assuming that any t'-time attacker $\mathcal{B}$ may make q_o queries to the *Plaintext Checking* (*PC*) *oracle* and $\mathcal{B}$'s advantage that finds a pre-image of a given ciphertext is less than ε'.

Theorem 1. *Suppose that the hash functions H_1, H_2, H_3 and H_4 are the random oracles. Then the proposed RMIBE scheme is a semantically outsider-secure RMIBE scheme (IND-O-sRMID-CCA) assuming that the Gap-BDH problem is hard. Concretely, assume that there is an outsider adversary $\mathcal{A}$ that has an advantage ε against the proposed RMIBE scheme. Suppose $\mathcal{A}$ makes at most $q_E > 0$ initial key extract queries, $q_U > 0$ time key update queries, $q_d > 0$ decryption queries and $q_i > 0$ queries to hash functions H_i $(i = 1, 2, 3, 4)$. Here we denote $q_o = q_3 + q_4$ (PC oracle queries). Then the proposed RMIBE scheme is $(t, q_1, q_2, q_3, q_4, q_E, q_U, q_d, \varepsilon)$-IND-O-sMRID-CCA secure assuming that the Gap-BDH is (t', q_o, ε')-intractable, where $\varepsilon' > \varepsilon - \dfrac{q_d}{2^y}$ and $t' < t + (q_1 + q_2 + q_E + q_U) \cdot O(\tau_1) + q_d \cdot O(\tau_2) + (q_3 + q_4) \cdot O(1)$, where τ_1 and τ_2 denote the executing time for a multiplication in G_1 and a pairing computation, respectively.*

▼**Proof.** Assume that an adversary $\mathcal{A}$ can break the proposed RMIBE scheme in the IND-sMRID-CCA game. By using the adversary $\mathcal{A}$, we may construct an OW-PCA adversary $\mathcal{B}$ to break the Bilinear ElGamal scheme. We assume that challenger $\mathcal{B}$ is given $\{G_1, G_2, \hat{e}, P, Q, P_{pub}\}$ as public keys of the Bilinear ElGamal scheme and $(U^*, W^*) = (r^*P, \hat{e}(Q, P_{pub})^{r^*} \cdot R^*)$ as a target ciphertext of the Bilinear ElGamal scheme. Suppose that $\mathcal{B}$ makes $q_o = q_3 + q_4$ queries to the *PC oracle* of the Bilinear ElGamal scheme within time t'. We denote B's winning probability by ε'. $\mathcal{B}$ simulates the challenger in IND-sMRID-CCA game for $\mathcal{A}$ as follows.

- *Phase* 1. $\mathcal{A}$ outputs target multiple identities denoted by $(ID_1^*, ID_2^*, \ldots, ID_n^*)$ and a target time period denoted by i^*.
- *Setup*: The challenger $\mathcal{B}$ creates the RMIBE public parameters $Parms = \{G_1, G_2, \hat{e}, P, P_{pub}, Q, H_1, H_2, H_3, H_4\}$ by setting $P_{pub} = bP$ and $Q = cP$. Then the challenger $\mathcal{B}$ gives $\mathcal{A}$ the public parameters $Parms$. Here H_1, H_2, H_3 and H_4 are random oracles controlled by $\mathcal{B}$. The challenger $\mathcal{B}$ answers queries issued by $\mathcal{A}$ as shown below.

– H_1 queries (ID): When $\mathcal{A}$ queries the oracle H_1 with ID, the challenger $\mathcal{B}$ performs the following tasks.
  (1) $\mathcal{B}$ maintains a list of tuples < ID, Q_{ID}, u > called the H_{list}^1.
  (2) If the query ID already appears on the H_{list}^1 in a tuple < ID, Q_{ID}, u >, then $\mathcal{B}$ responds with $H_1(ID) = Q_{ID}$.
  (3) Otherwise, $\mathcal{B}$ selects a random $u \in Z_q^*$ and computes Q_{ID} as follows:

$$Q_{ID} = H_1(ID) = \begin{cases} uP - Q \in G_1 & \text{if } ID = ID_j^* \text{ for } j \in [1,n], \\ uP \in G_1 & \text{if } ID \neq ID_j^* \text{ for } j \in [1,n]. \end{cases}$$

  Then $\mathcal{B}$ adds the tuple < ID, Q_{ID}, u > to the H_{list}^1. It responds to $\mathcal{A}$ with $H_1(ID) = Q_{ID}$.
– H_2 queries (ID, i): When $\mathcal{A}$ queries the oracle H_2 with (ID, i), the challenger $\mathcal{B}$ performs the following tasks.
  (1) $\mathcal{B}$ maintains a list of tuples < (ID, i), $R_{ID,i}$, v > called the H_{list}^2.
  (2) If the query (ID, i) already appears on the H_{list}^2 in a tuple < (ID, i), $R_{ID,i}$, v >, then $\mathcal{B}$ responds with $H_2(ID, i) = R_{ID,i}$.
  (3) Otherwise, $\mathcal{B}$ randomly selects a value $v \in Z_q^*$ and computes $R_{ID,i} = vP$. Then $\mathcal{B}$ adds the tuple < (ID, i), $R_{ID,i}$, v > to the H_{list}^2. It responds to $\mathcal{A}$ with $H_2(ID, i) = R_{ID,i}$.
– H_3 queries (R): When $\mathcal{A}$ queries the oracle H_3 with R, the challenger $\mathcal{B}$ performs the following tasks. Note that this hash function is related with the *PC oracle*.
  (1) $\mathcal{B}$ maintains a list of tuples < R, K > called the H_{list}^3.
  (2) If the query (R) already appears on the H_{list}^3 in a tuple <R, K>, then $\mathcal{B}$ responds with $H_3(R)$=K.
  (3) Otherwise, $\mathcal{B}$ checks whether (U^*, W^*) encrypts R using the *PC* oracle. If it is, it means that $\mathcal{B}$ finds out the correct message of (U^*, W^*). $\mathcal{B}$ returns R and terminates the game. If it is not, $\mathcal{B}$ randomly selects $K \in \{0, 1\}^x$. Then $\mathcal{B}$ adds < R, K > to the H_{list}^3. It responds to $\mathcal{A}$ with $H_3(R)$=K.
– H_4 queries (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L): $\mathcal{A}$ may issue queries with (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L) to H_4. $\mathcal{B}$ performs the following tasks.
  (1) $\mathcal{B}$ maintains a list of tuples (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L) called the H_{list}^4.
  (2) If the query (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L) already appears on the H_{list}^4 in a tuple < (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ >, then $\mathcal{B}$ responds with $H_4(R, m, U, V_1, V_2,\ldots, V_n, W_1, W_2, L) = \sigma$.
  (3) Otherwise, $\mathcal{B}$ checks whether (U^*, W^*) encrypts R using the *PC oracle*. If it is, it means that algorithm $\mathcal{B}$ finds out the correct message of (U^*, W^*). $\mathcal{B}$ returns R and terminates the game. If it is not, $\mathcal{B}$ randomly selects $\sigma \in \{0, 1\}^y$. Then $\mathcal{B}$ adds the tuple <(R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > to H_{list}^4. It responds to $\mathcal{A}$ with $H_4(R, m, U, V_1, V_2,\ldots, V_n, W_1, W_2, L) = \sigma$.

- *Phase* 2: Upon receiving the initial key extract query with ID and the time key update query with (ID, i), the challenger $\mathcal{B}$ responds as follows. Note that the associated hash queries have been asked before these queries.

– The initial key extract query (ID): Upon receiving this query with ID, the challenger $\mathcal{B}$ performs the following tasks.
  (1) Access the corresponding tuple < ID, Q_{ID}, u > from the list H_{list}^1.
  (2) Compute $D_{ID} = uP_{pub} \in G_1$.
  Observe that $D_{ID} = uP_{pub}=ubP=bQ_{ID}$, therefore D_{ID} is the initial secret key associated to the identity ID. Give D_{ID} to the adversary $\mathcal{A}$. The restriction here is that ID_j^* did not appear in this phase where j = 1, 2,…, n.
– The time key update query (ID, i): Upon receiving this query with (ID, i), the challenger $\mathcal{B}$ performs the following tasks.
  (1) Access the corresponding tuple < (ID, i), $R_{ID,i}$, v > from the list H_{list}^2.
  (2) Compute $T_{ID,i} = vP_{pub} \in G_1$.
  Observe that $T_{ID,i} = vP_{pub}= vbP = bR_{ID,i}$ and therefore $T_{ID,i}$ is the time update key associated to the identity ID and the period time i. Give $T_{ID,i}$ to $\mathcal{A}$.
– The decryption query ((ID^*, i^*), C): Upon receiving this query with ((ID^*, i^*), C), where C = (U, V_1, V_2,…, V_n, W_1, W_2, L, σ), the challenger $\mathcal{B}$ accesses the corresponding tuple <(R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > from the list H_{list}^4. Then $\mathcal{B}$ performs the following tasks.
  (1) If the tuple < (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > exists in the H_{list}^4, then $\mathcal{B}$ computes $H_3(R)$ using the simulation of H_3 above and checks whether $H_3(R) \oplus m = W_2$. If not, it returns "*Reject*", otherwise checks whether (U, W_1) encrypts R using the *PC oracle* and checks $\hat{e}(U, H_1(ID_j^*) + H_2(ID_j^*, i^*) + Q) = \hat{e}(P, V_j)$. If both of the equations hold, $\mathcal{B}$ returns m, otherwise returns "*Reject*".
  (2) If the tuple < (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > does not exist in the H_{list}^4, then $\mathcal{B}$ returns "*Reject*".

- *Challenge*: The adversary $\mathcal{A}$ outputs m_0^* and m_1^* on which it wishes to be challenged. The challenger $\mathcal{B}$ performs the following procedure.
  (1) Choose $\beta \in \{0,1\}$ and access the tuple < ID^*, Q_{ID^*}, u^* > from the list H_{list}^1 and the tuple < (ID^*, i^*), R_{ID^*,i^*}, v^* > from the list H_{list}^2 to get u_j^* and v_j^* respectively, for j = 1, 2,…, n.
  (2) Use the target ciphertext (U^*, W^*) = (r^*P, $\hat{e}(Q, P_{pub})^{r^*} \cdot R^*$) to compute $u_j^* \cdot U^*$ and $v_j^* \cdot U^*$ for j = 1, 2,…, n.

(3) Choose $K^* \in \{0, 1\}^x$ and $\sigma^* \in \{0, 1\}^y$ uniformly at random.

(4) Set $K^* = H_3(R^*)$ and $\sigma^* = H_4(R^*, m_\beta^*, U^*, (u_1^* \cdot U^* + v_1^* \cdot U^*), (u_2^* \cdot U^* + v_1^* \cdot U^*), \ldots, (u_n^* \cdot U^* + v_n^* \cdot U^*), W^*, K^* \oplus m_\beta^*, L^*)$, where L^* is created by $\mathcal{B}$.

(5) Define $C^* = (U^*, (u_1^* \cdot U^* + v_1^* \cdot U^*), (u_2^* \cdot U^* + v_1^* \cdot U^*), \ldots, (u_n^* \cdot U^* + v_n^* \cdot U^*), W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$. $\mathcal{B}$ gives C^* as the challenge to $\mathcal{A}$.

- Phase 3: The adversary $\mathcal{A}$ may issue more queries as in Phase 2.
- Guess: The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0.1\}$, and wins this game if $\beta' = \beta$.

In *Setup* and *Phase* 2, it is obvious that the challenger $\mathcal{B}$ perfectly simulates the random oracle H_1, H_2, H_3, H_4, the initial secret key extraction, the time key update and the decryption queries. The simulation of the ciphertext C^* is as follows:

$$C^* = (U^*, u_j^* \cdot U^* + v_j^* \cdot U^*, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, u_j^* \cdot r^* P + v_j^* \cdot r^* P, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, u_j^* \cdot r^* P - r^* Q + r^* Q + v_j^* \cdot r^* P, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, u_j^* \cdot r^* (u_j^* P - Q) + r^* Q + v_j^* \cdot r^* P, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, u_j^* \cdot r^* H_1(ID_j^*) + r^* H_2(ID_j^*, i_j^*) + r^* Q, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*),$$

for $j = 1, 2, \ldots, n$. Hence we know that C^* is a valid ciphertext.

Here, we analyze the algorithm $\mathcal{A}$'s advantage. If $\mathcal{A}$ has guessed a correct value σ without querying the random oracle H_4, algorithm $\mathcal{B}$ must terminate this simulation. If this situation may happen, the probability is $\frac{1}{2^y}$. Since in Phase 2, $\mathcal{A}$ makes total q_d decryption queries, we have the $\Pr[\text{Guess}H_4] \le \frac{q_d}{2^y}$, where $\text{Guess}H_4$ is the event which $\mathcal{A}$ guesses the correct value σ. If $W^*/R^* = \hat{e}(cP, bP)^a$, we have $\Pr[\mathcal{B}(P, aP, bP, cP) = \hat{e}(P, P)^{abc}] = |\Pr[\beta' = \beta \mid \neg \text{Guess}H_4] - 1/2|$ and $|\Pr[\beta' = \beta] - 1/2| > \varepsilon$. Consequently, we have $|\Pr[\beta' = \beta \mid \neg \text{Guess}H_4] - 1/2| > |\Pr[\beta' = \beta] - \Pr[\text{Guess}H_4] - 1/2| > \varepsilon - \Pr[\text{Guess}H_4] > \varepsilon - \frac{q_d}{2^y}$.

According to the above descriptions for the challenger $\mathcal{B}$, it is obvious that the required executing time for each H_1, H_2, initial key extract and time key update queries needs one multiplication computation in G_1. Performing q_d decryption queries requires q_d pairing computations. H_3 and H_4 queries need the time of performing *PC oracle*. So we have $t' < t + (q_1 + q_2 + q_E + q_U) \cdot O(\tau_1) + q_d \cdot O(\tau_2) + (q_3 + q_4) \cdot O(1)$, where τ_1 and τ_2 denote the executing time for a multiplication in G_1 and a pairing computation, respectively. ▲

In the following, we prove that our proposed RMIBE scheme is also a semantically insider-secure RMIBE scheme. Since the PKG stops to issue the current time update key for the revoked user, the user is unable to obtain the time update key in the present time period. We give a theorem for an insider attacker (revoked user) and prove that insider adversary (or a revoked user) cannot decrypt the message.

Theorem 2. *Suppose that the hash functions H_1, H_2, H_3 and H_4 are the random oracles. Then the proposed RMIBE scheme is a semantically insider-secure RMIBE scheme (IND-I-sRMID-CCA) assuming that the Gap-BDH problem is hard. Concretely, assume that there is an insider adversary $\mathcal{A}$ that has an advantage ε against the proposed RMIBE scheme. Suppose $\mathcal{A}$ makes at most $q_E > 0$ initial key extract queries, $q_U > 0$ time key update queries, $q_d > 0$ decryption queries and $q_i > 0$ queries to hash functions H_i ($i = 1, 2, 3, 4$). Here we denote $q_o = q_3 + q_4$ (PC oracle queries). Then the proposed RMIBE scheme is $(t, q_1, q_2, q_3, q_4, q_E, q_U, q_d, \varepsilon)$-IND-O-sMRID-CCA secure assuming that the Gap-BDH is (t', q_o, ε')-intractable, where $\varepsilon' > \varepsilon - \frac{q_d}{2^y}$ and $t' < t + (q_1 + q_2 + q_E + q_U) \cdot O(\tau_1) + q_d \cdot O(\tau_2) + (q_3 + q_4) \cdot O(1)$, where τ_1 and τ_2 denote the executing time for a multiplication in G_1 and a pairing computation, respectively.*

▼***Proof.*** Assume that an adversary $\mathcal{A}$ can break the proposed RMIBE scheme in the IND-sMRID-CCA game. By using the adversary $\mathcal{A}$, we may construct an OW-PCA adversary $\mathcal{B}$ to break the Bilinear ElGamal scheme. We assume that challenger $\mathcal{B}$ is given $\{G_1, G_2, \hat{e}, P, Q, P_{pub}\}$ as public keys of the Bilinear ElGamal scheme and $(U^*, W^*) = (r^* P, \hat{e}(Q, P_{pub})^{r^*} \cdot R^*)$ as a t arget ciphertext of the Bilinear ElGamal scheme. Suppose that $\mathcal{B}$ makes $q_o = q_3 + q_4$ queries to the *PC oracle* of the Bilinear ElGamal scheme within time t'. We denote B's winning probability by ε'. $\mathcal{B}$ simulates the challenger in IND-sMRID-CCA game for $\mathcal{A}$ as follows.

- *Phase* 1. $\mathcal{A}$ outputs target multiple identities denoted by $(ID_1^*, ID_2^*, \ldots, ID_n^*)$ and a target time period denoted by i^*.
- *Setup*: The challenger $\mathcal{B}$ creates the RMIBE public parameters $Parms = \{G_1, G_2, \hat{e}, P, P_{pub}, Q, H_1, H_2, H_3, H_4\}$ by setting $P_{pub} = bP$ and $Q = cP$. Then the challenger $\mathcal{B}$ gives $\mathcal{A}$ the public parameters *Parms*. Here H_1, H_2, H_3 and H_4 are random oracles controlled by $\mathcal{B}$. The challenger $\mathcal{B}$ answers queries issued by $\mathcal{A}$ as shown below.

– H_1 queries (ID): When $\mathcal{A}$ queries the oracle H_1 with ID, the challenger $\mathcal{B}$ performs the following tasks.
  (1) $\mathcal{B}$ maintains a list of tuples < ID, Q_{ID}, v > called the H_{list}^1.
  (2) If the query ID already appears on the H_{list}^1 in a tuple < ID, Q_{ID}, v >, then $\mathcal{B}$ responds with $H_1(ID) = Q_{ID}$.
  (3) Otherwise, $\mathcal{B}$ randomly selects a value $v \in Z_q^*$ and computes $Q_{ID} = vP$. Then $\mathcal{B}$ adds the tuple < ID, Q_{ID}, v > to the H_{list}^1. It responds to $\mathcal{A}$ with $H_1(ID) = Q_{ID}$.
– H_2 queries (ID, i): When $\mathcal{A}$ queries the oracle H_2 with (ID, i), the challenger $\mathcal{B}$ performs the following tasks.
  (1) $\mathcal{B}$ maintains a list of tuples < (ID, i), $R_{ID,i}$, u > called the H_{list}^2.
  (2) If the query (ID, i) already appears on the H_{list}^2 in a tuple < (ID, i), $R_{ID,i}$, u >, then $\mathcal{B}$ responds with $H_2(ID, i) = R_{ID,i}$.
  (3) Otherwise, $\mathcal{B}$ selects a random $u \in Z_q^*$ and computes Q_{ID} as follows:

$$R_{ID,i} = H_1(ID,i) = \begin{cases} uP - Q \in G_1 & \text{if } (ID,i) = (ID_j^*, i_j^*) \text{ for } j \in [1,n], \\ uP \in G_1 & \text{if } (ID,i) \neq (ID_j^*, i_j^*) \text{ for } j \in [1,n]. \end{cases}$$

Then $\mathcal{B}$ adds the tuple < (ID, i), $R_{ID,i}$, u > to the H_{list}^2. It responds to $\mathcal{A}$ with $H_2(ID, i) = R_{ID,i}$.
– H_3 queries (R): When $\mathcal{A}$ queries the oracle H_3 with R, the challenger $\mathcal{B}$ performs the following tasks. Note that this hash function is related with the *PC oracle*.
  (1) $\mathcal{B}$ maintains a list of tuples < R, K > called the H_{list}^3.
  (2) If the query (R) already appears on the H_{list}^3 in a tuple <R, K>, then $\mathcal{B}$ responds with $H_3(R)$=K.
  (3) Otherwise, $\mathcal{B}$ checks whether (U^*, W^*) encrypts R using the *PC oracle*. If it is, it means that $\mathcal{B}$ finds out the correct message of (U^*, W^*). $\mathcal{B}$ returns R and terminates the game. If it is not, $\mathcal{B}$ randomly selects $K \in \{0, 1\}^x$. Then $\mathcal{B}$ adds < R, K > to the H_{list}^3. It responds to $\mathcal{A}$ with $H_3(R)$=K.
– H_4 queries (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L): $\mathcal{A}$ may issue queries with (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L) to H_4. $\mathcal{B}$ performs the following tasks.
  (1) $\mathcal{B}$ maintains a list of tuples (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L) called the H_{list}^4.
  (2) If the query (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L) already appears on the H_{list}^4 in a tuple < (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ >, then $\mathcal{B}$ responds with $H_4(R, m, U, V_1, V_2,\ldots, V_n, W_1, W_2, L) = \sigma$.
  (3) Otherwise, $\mathcal{B}$ checks whether (U^*, W^*) encrypts R using the *PC oracle*. If it is, it means that algorithm $\mathcal{B}$ finds out the correct message of (U^*, W^*). $\mathcal{B}$ returns R and terminates the game. If it is not, $\mathcal{B}$ randomly selects $\sigma \in \{0, 1\}^y$. Then $\mathcal{B}$ adds the tuple <(R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > to H_{list}^4. It responds to $\mathcal{A}$ with $H_4(R, m, U, V_1, V_2,\ldots, V_n, W_1, W_2, L) = \sigma$.

- *Phase* 2: Upon receiving the initial key extract query with ID and the time key update query with (ID, i), the challenger $\mathcal{B}$ responds as follows. Note that the associated hash queries have been asked before these queries.

– The initial key extract query (ID): Upon receiving this query with ID, the challenger $\mathcal{B}$ performs the following tasks.
  (1) Access the corresponding tuple < ID, Q_{ID}, v > from the list H_{list}^1.
  (2) Compute $D_{ID} = vP_{pub} \in G_1$.
  Observe that $D_{ID} = vP_{pub}=vbP=bQ_{ID}$, therefore D_{ID} is the initial secret key associated to the identity ID. Give D_{ID} to the adversary $\mathcal{A}$. The restriction here is that ID_j^* did not appear in this phase where j = 1, 2,…, n.
– The time key update query (ID, i): Upon receiving this query with (ID, i), the challenger $\mathcal{B}$ performs the following tasks.
  (1) Access the corresponding tuple < (ID, i), $R_{ID,i}$, u > from the list H_{list}^2.
  (2) Compute $T_{ID,i} = uP_{pub} \in G_1$.
  Observe that $T_{ID,i} = uP_{pub}= ubP = uR_{ID,i}$ and therefore $T_{ID,i}$ is the time update key associated to the identity ID and the period time i. Give $T_{ID,i}$ to $\mathcal{A}$.
– The decryption query ((ID^*, i^*), C): Upon receiving this query with ((ID^*, i^*), C), where C = (U, V_1, V_2,…, V_n, W_1, W_2, L, σ), the challenger $\mathcal{B}$ accesses the corresponding tuple <(R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > from the list H_{list}^4. Then $\mathcal{B}$ performs the following tasks.
  (1) If the tuple < (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > exists in the H_{list}^4, then $\mathcal{B}$ computes $H_3(R)$ using the simulation of H_3 above and checks whether $H_3(R) \oplus m = W_2$. If not, it returns "*Reject*", otherwise checks whether (U, W_1) encrypts R using the *PC oracle* and checks $\hat{e}(U, H_1(ID_j^*) + H_2(ID_j^*, i^*) + Q) = \hat{e}(P, V_j)$. If both of the equations hold, $\mathcal{B}$ returns m, otherwise returns "*Reject*".
  (2) If the tuple < (R, m, U, V_1, V_2,…, V_n, W_1, W_2, L), σ > does not exist in the H_{list}^4, then $\mathcal{B}$ returns "*Reject*".

- *Challenge*: The adversary $\mathcal{A}$ outputs m_0^* and m_1^* on which it wishes to be challenged. The challenger $\mathcal{B}$ performs the following procedure.
  (1) Choose $\beta \in \{0,1\}$ and access the tuple < ID^*, Q_{ID^*}, v^* > from the list H_{list}^1 and the tuple < (ID^*, i^*), R_{ID^*,i^*}, u^* > from the list H_{list}^2 to get v_j^* and u_j^* respectively, for j = 1, 2,…, n.
  (2) Use the target ciphertext (U^*, W^*) = (r^*P, $\hat{e}(Q, P_{pub})^{r^*} \cdot R^*$) to compute $v_j^* \cdot U^*$ and $u_j^* \cdot U^*$ for j = 1, 2,…, n.
  (3) Choose $K^* \in \{0, 1\}^x$ and $\sigma^* \in \{0, 1\}^y$ uniformly at random.

(4) Set $K^* = H_3(R^*)$ and $\sigma^* = H_4(R^*, m_\beta^*, U^*, (v_1^* \cdot U^* + u_1^* \cdot U^*), (v_2^* \cdot U^* + u_2^* \cdot U^*), \ldots, (v_n^* \cdot U^* + u_n^* \cdot U^*), W^*, K^* \oplus m_\beta^*, L^*)$, where L^* is created by $\mathcal{B}$.

(5) Define $C^* = (U^*, (v_1^* \cdot U^* + u_1^* \cdot U^*), (v_2^* \cdot U^* + u_2^* \cdot U^*), \ldots, (v_n^* \cdot U^* + u_n^* \cdot U^*), W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$. $\mathcal{B}$ gives C^* as the challenge to $\mathcal{A}$.

- *Phase* 3: The adversary $\mathcal{A}$ may issue more queries as in *Phase* 2.
- *Guess:* The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0.1\}$, and wins this game if $\beta' = \beta$.

In *Setup* and *Phase* 2, it is obvious that the challenger $\mathcal{B}$ perfectly simulates the random oracle H_1, H_2, H_3, H_4, the initial secret key extraction, the time key update and the decryption queries. The simulation of the ciphertext C^* is as follows:

$$C^* = (U^*, v_j^* \cdot U^* + u_j^* \cdot U^*, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, v_j^* \cdot r^* P + u_j^* \cdot r^* P, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, v_j^* \cdot r^* P - r^* Q + r^* Q + u_j^* \cdot r^* P, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, v_j^* \cdot r^* (v_j^* P - Q) + r^* Q + u_j^* \cdot r^* P, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*)$$
$$= (U^*, v_j^* \cdot r^* H_1(ID_j^*) + r^* H_2(ID_j^*, i_j^*) + r^* Q, W^*, K^* \oplus m_\beta^*, L^*, \sigma^*),$$

for $j = 1, 2, \ldots, n$. Hence we know that C^* is a valid ciphertext.

The analysis is similar to Theorem 1. The successful probability (advantage) of the challenger $\mathcal{B}$ who can solve the CDH problem is at least $\varepsilon - \frac{q_d}{2^y}$. The executing time is $t + (q_1 + q_2 + q_E + q_U) \cdot O(\tau_1) + q_d \cdot O(\tau_2) + (q_3 + q_4) \cdot O(1)$, where τ_1 and τ_2 denote the executing time for a multiplication in G_1 and a pairing computation, respectively.▲

6. Performance analysis and comparisons

In this section, we analyze the performance of the proposed RMIBE scheme and give the comparisons with the Tseng-Tsai RIBE scheme [18]. For convenience, we define the following notations to analyze the computational cost.

- TG_e: The time of executing a bilinear pairing operation $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$.
- TG_{mul}: The time of executing a multiplication operation in G_1.
- T_{exp}: The time of executing an exponentiation operation in G_2.
- TG_H: The time of executing a map-to-point hash function $H_1(\,)$ or $H_2(\,)$.

Some simulation results in [27-30] demonstrate that executing a bilinear pairing operation TG_e is more time-consuming than other operations. In order to obtain more precise analysis of the encryption cost for n receivers, we use the simulation results in [30] to evaluate it. Table 1 lists the simulation results of TG_e, T_{exp} and TG_H with respect to TG_{mul}, respectively. The simulation environment is presented as follows. The processor is an Intel Core Duo T2400 1.83GHz with 3 GB of RAM 533 MHz. The cryptographic pairing system uses Weil bilinear pairing system in which the used pairing values belong to a finite field of 1024 bits. The computation costs of TG_e, TG_H and T_{exp} are equal to about $9TG_{mul}$, $0.7TG_{mul}$ and $1TG_{mul}$, respectively.

Table 1. The cost of the related pairing based operations

	TG_e	TG_H	T_{exp}
Cost	$9TG_{mul}$	$0.7TG_{mul}$	$1TG_{mul}$

In the following, we analyze the computational costs of the proposed RMIBE scheme. For encrypting a message for n receivers in the proposed RMIBE scheme, it takes $TG_e + (n+1) \cdot TG_{mul} + 2n \cdot TG_H + T_{exp}$ time. For each selected receiver's decryption in the proposed RMIBE scheme, it requires $2TG_e + T_{exp}$ time. Table 2 lists the comparisons between the proposed RMIBE scheme and the Tseng-Tsai RIBE scheme [18] in terms of the computational costs of encryption/decryption for n receivers, and security assumption. Because Tseng and Tsai didn't address the problem where a sender would like to encrypt a message for n receivers, the sender must re-encrypt the message n times using their RIBE scheme. Thus, a sender needs n expensive pairing operations to encrypt a single message for n receivers. Our proposed scheme requires only one pairing operation to encrypt a message for n receivers. The required computational cots of encryption for n users are depicted in Fig. 1. It is obvious that our MIBE scheme is better than the Tseng-Tsai RIBE scheme for encrypting a message for n receivers.

Table 2. Comparison between the proposed scheme and the Tseng-Tsai IBE scheme.

	The Tseng-Tsai's RIBE scheme [18]	**Our proposed RMIBE scheme**
Security assumption	BDH assumption	Gap-BDH assumption
Computational cost of encryption for n receivers	$n \cdot TG_e + TG_{mul} + 3n \cdot TG_H + n \cdot T_{exp}$	$TG_e + (n+1) \cdot TG_{mul} + 2n \cdot TG_H + T_{exp}$
$n = 50$	606 TG_{mul}	131 TG_{mul}
$n = 100$	1211 TG_{mul}	251 TG_{mul}
$n = 200$	2421 TG_{mul}	491 TG_{mul}
$n = 500$	6051 TG_{mul}	1211 TG_{mul}
Computational cost of decryption for each user	$2TG_e + TG_{mul} + T_{exp}$	$2TG_e + T_{exp}$

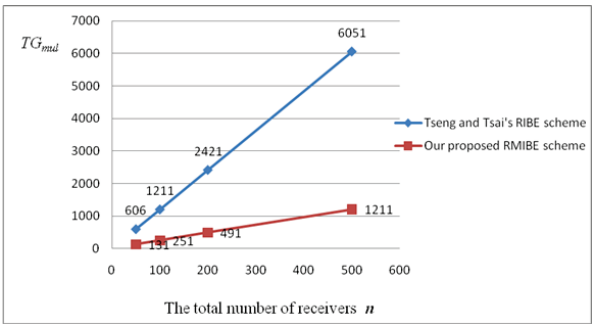

Figure 1. Performance comparison of encryption for n users

7. Conclusions

In this paper, we addressed the problem where one sender encrypts an identical message for multiple receivers in the revocable ID-based version. We defined the framework of revocable multi-receiver ID-based encryption (RMIBE) with a p ublic channel. Meanwhile, the security notions were completely defined to formalize the possible threats and attacks that include an outside adversary and a revoked user (an inside adversary). We proposed the concrete RIBE scheme from bilinear pairings. Based on the Gap-bilinear Diffie-Hellman assumption, the RMIBE scheme is proved to be semantically secure against adaptive chosen ciphertext attacks (CCA) under the selective-ID version. Performance analysis and comparisons are made to demonstrate that our proposed scheme requires only one pairing operation for encrypting a message for multiple identities and provides revocable property.

Acknowledgements

This research was partially supported by National Science Council, Taiwan, R.O.C., under contract no. NSC101-2221-E-018-027.

References

[1] **A. Shamir.** Identity-based cryptosystems and signature schemes. In: *Proc. of CRYPTO'84*, LNCS, 196, 1984, pp. 47–53.

[2] **D. Boneh, M. Franklin.** Identity-based encryption from the Weil pairing. In: *Proc. of CRYPTO'01*, LNCS, 2139, 2001, pp. 213–229.

[3] **J. C. Cha, J. H. Cheon.** An identity-based signature from gap Diffie-Hellman groups. In: *Proc. of PKC'03*, LNCS, 2567, 2003, pp. 18–30.

[4] **M. Bellare, C. Namprempre, G. Neven**. Security proofs for identity-based identification and signature schemes. *Journal of Cryptology*, 2009, Vol. 22, No. 1, pp. 1–61.

[5] **B. Waters.** Efficient identity-based encryption without random oracles. In: *Proceedings of Eurocrypt'05*, LNCS, 2005, Vol. 3494, pp. 114–127.

[6] **L. Chen, Z. Cheng, N.P. Smart.** Identity-based key agreement protocols from pairings. *International Journal of Information Security*, 2007, Vol. 6, No. 4, pp. 213–241.

[7] **D. Boneh, M. Hamburg.** Generalized identity based and broadcast encryption schemes. In: *Proc. of Asiacrypt'08*, LNCS, 5350, 2008, pp. 455-470.

[8] **Y. H. Chuang, Y. M. Tseng.** Towards generalized ID-based user authentication for mobile multi-server environment. *International Journal of Communication Systems*, 2012, Vol. 25, No. 4, pp. 447-460.

[9] **Y. F. Chang, W. L. Tai, C. Y. Lin.** A verifiable proxy signature scheme based on bilinear pairings with identity-based cryptographic approaches. *Information Technology and Control*, 2012, Vol. 41, No. 1, pp. 60–68.

[10] **T. Y. Wu, Y. M. Tseng.** An ID-based mutual authentication and key exchange protocol for low-power mobile devices. *The Computer Journal*, 2010, Vol. 53, No. 7, pp. 1062–1070.

[11] **E. J. Yoon.** An efficient and secure identity-based strong designated verifier signature scheme. *Information Technology and Control*, 2011, Vol. 40, No. 4, pp. 323–329.

[12] **T. Y. Wu, Y. M. Tseng.** Towards ID-based authenticated group key exchange protocol with identifying malicious participants. *Informatica*, 2012, Vol. 23, No. 2, pp. 315–334.

[13] **T. Y. Wu, Y. M. Tseng, T. T. Tsai.** A revocable ID-based authenticated group key exchange protocol with resistant to malicious participants. *Computer Networks*, 2012, 56 (12), 2994-3006.

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

[15] **A. Boldyreva, V. Goyal, V. Kumar.** Identity-based encryption with efficient revocation. In: *Proceedings of CCS'08*, 2008, pp. 417–426.

[16] **R. Canetti, S. Halevi, J. Katz.** A forward-secure public-key encryption scheme. In: *Proceedings of Eurocrypt'03*, LNCS, 2656, 2003, pp. 255–271.

[17] **B. Libert, D. Vergnaud.** Adaptive-ID secure revocable identity-based encryption. In: *Proceedings of CT-RSA'09*, LNCS, 5473, 2009, pp. 1–15.

[18] **Y. M. Tseng, T. T. Tsai.** Efficient revocable ID-based encryption with a public channel. *The Computer Journal*, 2012, Vol. 55, No. 4, pp. 475–486.

[19] **D. Boneh, M. Franklin.** Identity-based encryption from the Weil pairing. *SIAM J. of Computing*, 2003, Vol. 32, No. 3, pp. 586–615.

[20] **J. Baek, R. Safavi-Naini, W. Susilo.** Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In: *Proceedings of PKC'05*, LNCS, 3386, 2005, pp. 380–397.

[21] **J. Baek, Y. Zheng.** Identity-based threshold decryption. In: *Proceedings of PKC'04*, LNCS, 2947, 2004, pp. 262–276.

[22] **R. Sakai, M. Kasahara.** ID-based cryptosystems with pairing on elliptic curve. In: *Cryptology ePrint Archive*, 2003, Report 2003/054.

[23] **S. Galbraith, K. Paterson, N.P. Smart.** Pairings for cryptographers. *Discrete Applied Mathematics*, 2008, Vol. 156, No. 16, pp. 3113–3121.

[24] **T. Okamoto, D. Pointcheval.** REACT: Rapid enhanced-security asymmetric cryptosystem transform. In: *Proceedings of CT-RSA'01*, 2001, pp. 159–174.

[25] **M. Bellare, P. Rogaway.** Random oracles are practical: a paradigm for designing efficient protocols. In: *Proceedings of CCS'93*, 1993, pp. 62–73.

[26] **R. Canetti, O. Goldreich, S. Halevi.** The random oracle methodology, revisited. *Journal of ACM*, 2004, Vol. 51, No. 4, pp. 557–594.

[27] **M. Scott.** Computing the Tate pairing. In: *Proceedings of CT-RSA'05*, LNCS, 3376, 2005, pp. 293–304.

[28] **X. Cao, X. Zeng, W. Kou, L. Hu.** Identity-based anonymous remote authentication for value-added services in mobile networks. In: *IEEE Trans. Veh. Technol.*, 2009, Vol. 58, No. 7, pp. 3508–3517.

[29] **T. Y. Wu, Y. M. Tseng.** An efficient user authentication and key exchange protocol for mobile client-server environment. *Computer Networks*, 2010, Vol. 54, No. 9, pp. 1520–1530.

[30] **G. Stephanides.** Short-key certificateless encryption. In: *Proceedings of LightSec'11*, 2011, pp. 69–75.

Received August 2012.