A Generic Construction of Revocable Identity-Based Encryption

Xuecheng Ma¹,² and Dongdai Lin¹,²

¹ State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
² School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
maxuecheng@iie.ac.cn

Abstract. Revocable identity-based encryption (RIBE) is an extension of IBE that supports a key revocation mechanism, which is important when deployed an IBE system in practice. Boneh and Franklin presented the first generic construction of RIBE, however, their scheme is not scalable where the size of key update is linear in the number of users in the system. Then, Boldyreva, Goyal and Kumar presented the first scalable RIBE where the size of key update is logarithmic in the number of users and linear in the number of revoked users.

In this paper, we present a generic construction of scalable RIBE from any IBE in a black-box way. Our construction has some merits both in theory and in practice. We obtain the first RIBE scheme based on quadratic residuosity problem and the first adaptively secure RIBE scheme based on lattices if we instantiate the underlying IBE with IBE schemes from quadratic residuosity assumption and adaptively secure IBE from lattices, respectively. In addition, the size of public parameters and secret keys are the same as that of the underlying IBE schemes. In server-aided model, the overheads of communication and computation for receivers are the same as those of underlying IBE schemes. Furthermore, the storage overhead for key update in our scheme is constant (in the number of users) while it was linear in the number of users in previous works.

Key words: Generic Construction, Revocable Identity Based Encryption
1 Introduction

Identity-Based Encryption (IBE) was introduced by Shamir [41], to eliminate the need for maintaining a certificate based Public Key Infrastructure (PKI) in the traditional Public Key Encryption (PKE) setting. The first IBE scheme was proposed by Boneh and Franklin [7] in the random oracle model [3]. Since then, realizations from bilinear maps [5, 6, 44, 20, 45], from quadratic residues modulo composite [14, 8], from lattices [1, 2, 9–11, 21, 46, 47] and from the computational Diffie-Hellman assumption [18] have been proposed.

Revocation capability is very important and necessary for IBE setting as well as PKI setting. Boneh and Franklin [7] proposed a naive method for adding a simple revocation mechanism to any IBE system as follows. A sender encrypts a message using a receiver’s identity concatenated with the current time period, i.e., id||t and the Key Generation Center (KGC) issues the private key sk_{id||t} for each non-revoked users in every time period. However, BF-RIBE scheme is inefficient. The number of private keys issued in every time period is linear in the number of all users in the system hence the scheme did not scale well if the number of users became too large.

Boldyreva, Goyal and Kumar (BGK) [4] proposed the first scalable revocable IBE (RIBE) scheme in the selective security model by combining the fuzzy IBE scheme of Sahai and Waters [38] with a subset cover framework called the complete subtree (CS) method [31]. The BGK scheme significantly reduced the size of key updates from linear to logarithmic in the number of users. Each user holds a long-term private key associated with its identity but the private key is not allowed to decrypt the ciphertext in order to achieve the key revocation mechanism. KGC broadcasts key updates for every time period through a public channel. Specially, the non-revoked users can derive decryption key from their long-term private keys and key updates while revoked users can’t. There are numerous followup works [24, 27, 29, 39, 43].

RIBE with DKER. In the definition of security in BGK-RIBE, the adversary is only allowed to be access to the key extraction oracle, the revocation oracle and the key update oracle. Considering leakage of decryption keys in realistic attacks, Seo and Emura [39, 40] introduced a security notion called decryption key exposure resistance (DKER). In the definition of DKER security experiment, an exposure of a user’s decryption key at some time period will not compromise the confidentiality of ciphertexts that are encrypted for different time periods. It
attracted many followup works concerning R(H)IBE schemes with DKER [19, 24, 26–28, 30, 33, 34, 37, 40, 43]. Recently, Katsumata et al. [25] presented a generic construction of RIBE with DKER from any RIBE without DKER and two-level HIBE. Combining the result of [17] that any IBE schemes can be converted to an HIBE scheme (in the selective-identity model) and any RIBE scheme without DKER implies an IBE scheme, their result also implies a generic conversion from any RIBE scheme without DKER into an RIBE scheme with DKER.

**Lattice-Based RIBE.** The first selectively-secure lattice-based RIBE without DKER was proposed by Chen et al. [12]. Cheng and Zhang [13] claimed that their proposed RIBE scheme with the subset difference (SD) method is the first adaptively secure lattice-based scheme. However, Takayasu and Watanabe [42] pointed out critical bugs in their security proof and presented a semi-adaptively secure lattice-based RIBE scheme with bounded DKER which only allows a bounded number of decryption keys to be leaked. Recently, Katsumata et al. [25] proposed the first lattice-based R(H)IBE scheme with DKER secure under the learning with errors (LWE) assumption but their proposal was still selectively secure. Therefore, constructing an adaptively secure RIBE scheme even without DKER based on lattices still remains an open problem.

**Server-aided RIBE** [35, 15, 32] is a variant of RIBE where almost all of the workload on the user side can be delegated to an untrusted third party server. The server is untrusted in the sense that it does not possess any secret information. Each user only need to store a short long-term private key without having to communicate with either KGC or the third party server.

**Our Contributions.** In this paper, we propose a generic construction of RIBE from any IBE schemes in a black-box way. The update key size of our construction is logarithmic in the number of users. The benefits of such a generic construction are as follows:

- **Practical Benefits.**
  
  (a) Our RIBE scheme has the same size of public parameters and user’s secret key as those of underlying IBE scheme. Although the size of ciphertext in our scheme is logarithmic in the number of users, fortunately, there is a tradeoff between the size of public parameter and size of the ciphertext if we replace the underlying IBE with appropriate Identity Based Broadcast Encryption (IBBE).
The storage overhead for key updates in our scheme is only constant in the number of users. Instead of storing information in every node of the binary tree in previous works, our construction leverages the master secret key of IBE to generate key updates. Due to compression of the master secret key, the KGC needs only constant storage for key updates in our construction.

(c) Our scheme is naturally server-aided. The communication cost and computation cost for the receiver is the same as the underlying IBE scheme in the server-aided model.

An overview comparing the efficiency of our revocable IBE scheme to those of other revocable IBE schemes is given in Table 1.

- Theoretical Benefits. There have been a lot of works considering ad hoc methods to transform existing IBE schemes with revocation mechanism. However, as the only generic construction, BF-RIBE is not scalable. Our generic construction demonstrates a simple and clear picture about how revocation problems in IBE could be addressed.

(a) We present a generic construction of RIBE that can convert any IBE schemes to RIBE schemes without DKER. Combining the conversion from RIBE without DKER to RIBE with DKER in [25], our result also implies a generic construction of RIBE with DKER from any IBE.

(b) Instantiating our generic construction of existing IBE schemes [14, 8], we can obtain the first RIBE schemes based on quadratic residues modulo composite.

(c) Our construction inherent the security of the underlying IBE scheme. Hence, we can obtain the first adaptively-secure lattice-based RIBE scheme by instantiating our construction with adaptively-secure IBE from lattices [1, 2, 9–11, 21, 46, 47].

**Related Work.** The first revocable IBE scheme from any IBE was presented by Boneh and Franklin [7], however their proposal was not scalable. Boldyreva et al. [4] proposed the first scalable RIBE but their scheme was not a generic construction. Recently, Katsumata et al. [25] proposed a generic construction of RIBE with DKER which uses as building blocks any two-level standard HIBE scheme and (weak) RIBE scheme without DKER.

Identity-Based Broadcast Encryption is a natural extension of IBE. Deleglise [16] presented the first IBBE scheme with constant size ciphertext and
Table 1. Comparison of revocable identity-based encryption schemes

<table>
<thead>
<tr>
<th>Schemes</th>
<th>BF</th>
<th>BGK</th>
<th>LV</th>
<th>SE</th>
<th>LLP</th>
<th>Ours-1</th>
<th>Ours-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP Size</td>
<td>O(1)</td>
<td>O(1)</td>
<td>O(λ)</td>
<td>O(λ)</td>
<td>O(N + λ)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>SK Size</td>
<td>O(log(N))</td>
<td>O(log(N))</td>
<td>O(log(N))</td>
<td>O(log^3 N)</td>
<td>O(1)</td>
<td>O(1)</td>
<td></td>
</tr>
<tr>
<td>CT Size</td>
<td>O(1)</td>
<td>O(1)</td>
<td>O(1)</td>
<td>O(1)</td>
<td>O(log(N))</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>KU Size</td>
<td>O(N − r)</td>
<td>O(r log N)</td>
<td>O(r log N)</td>
<td>O(r log N)</td>
<td>O(r log N)</td>
<td>O(r log N)</td>
<td>O(r log N)</td>
</tr>
<tr>
<td>DKER</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>Full</td>
<td>Selective</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td></td>
</tr>
<tr>
<td>Assumption</td>
<td>RO,BDH</td>
<td>DBDH</td>
<td>DBDH</td>
<td>DBDH</td>
<td>Static</td>
<td>RO,BDH</td>
<td>DBDH,Static</td>
</tr>
</tbody>
</table>

We let λ be a security parameter, N be the number of maximum users, r be the number of revoked users. For security model, we use symbols RO for random oracle model, Full for adaptive model, Selective for selective model. The storage is what KGC needs for key updates. Note that our two schemes are the result of combing our generic construction with the generic construction in [25]. In our-1, we instantiate the IBE scheme and HIBE scheme with [7] and [22] respectively. In our-2, we instantiate the IBBE scheme with constant-size ciphertext and secret key and two-level HIBE scheme with [48] and [44], respectively.

with weak selective security in the random oracle model. Gentry and Waters [23] were the first to propose adaptively secure IBBE systems achieving linear and sub-linear sized ciphertexts. Zhang et al. [48] presented an adaptively secure identity-based broadcast encryption scheme with a constant-size ciphertext and private keys. Recently, Ramanna [36] proposed a novel IBBE scheme with constant size ciphertext that can achieve adaptive security in the standard model.

2 Preliminaries

2.1 Notations

Throughout the paper we use the following notation: We use λ as the security parameter and write negl(λ) to denote that some function f(·) is negligible in λ. An algorithm is PPT if it is modeled as a probabilistic Turing machine whose running time is bounded by some function poly(λ). By X ≈ Y, we denote that the random variable ensembles \(\{X_\lambda\}_{\lambda \in \mathbb{N}}\) and \(\{Y_\lambda\}_{\lambda \in \mathbb{N}}\) are computationally indistinguishable with error negl(λ). If S is a finite set, then \(s \leftarrow S\) denotes the operation of picking an element s from S uniformly at random. If A is a probabilistic algorithm, then \(y \leftarrow A(x)\) denotes the action of running A(x) on input x with
uniform coins and outputting $y$. Let $[u]$ denotes $\{1, \ldots, n\}$. Let $\{0, 1\}^{[i,j]}$ denotes all binary strings with length in $[i, j]$. For a bit string $a = (a_1, \ldots, a_n) \in \{0, 1\}^n$, and $i, j \in [n]$ with $i \leq j$, we write $a[i, j]$ to denote the substring $(a_i, \ldots, a_j)$ of $a$. For any two strings $u$ and $v$, $|u|$ denote the length of $u$ and $u|v$ denotes their concatenation. Let $\mathcal{B}T$ be a complete binary tree and $\text{Path}(v)$ be a set of all nodes on the path between the root node and a leaf $v$. We also use $\text{Path}(id)$ to denote the path from the corresponding node of $id$ to the root node.

2.2 Identity-Based Encryption

An identity-based encryption scheme consists of four probabilistic polynomial-time (PPT) algorithms ($\text{Setup}$, $\text{KeyGen}$, $\text{Enc}$, $\text{Dec}$) defined as follows:

- $\text{Setup}(1^\lambda)$: This algorithm takes as input the security parameter $1^\lambda$, and outputs a public parameter $\mathcal{P}P$ and a master secret key $\mathcal{M}K$.
- $\text{KeyGen}(\mathcal{M}K, id)$: This algorithm takes as input the master secret key $\mathcal{M}K$ and an identity $id \in \{0, 1\}^\ell$, it outputs the identity secret key $sk_{id}$.
- $\text{Enc}(\mathcal{P}P, id, \mu)$: This algorithm takes as input the public parameter $\mathcal{P}P$, an identity $id \in \{0, 1\}^\ell$, and a plaintext $\mu$, it outputs a ciphertext $c$.
- $\text{Dec}(sk_{id}, c)$: This algorithm takes as input a secret key $sk_{id}$ for identity $id$ and a ciphertext $c$, it outputs a plaintext $\mu$.

The following completeness and security properties must be satisfied:

- **Completeness**: For all security parameters $1^\lambda$, identity $id \in \{0, 1\}^\ell$ and plaintext $\mu$, the following holds:
  
  $$\Pr[\text{Dec}(sk_{id}, \text{Enc}(\mathcal{P}P, id, \mu)) = \mu] = 1$$

  where $(\mathcal{P}P, \mathcal{M}K) \leftarrow \text{Setup}(1^\lambda)$ and $sk_{id} \leftarrow \text{KeyGen}(\mathcal{M}K, id)$.

- **Selective Security**: For any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, there is a negligible function $\text{negl}(\cdot)$ such that the advantage of $\mathcal{A}$ satisfies:
  
  $$\text{Adv}^{\text{IND-sID-CPA}}_{\mathcal{A}} = | \Pr[\text{IND-sID-CPA}(\mathcal{A}) = 1] - \frac{1}{2} | \leq \text{negl}(\lambda)$$

  where $\text{IND-sID-CPA}(\mathcal{A})$ is shown in Figure 1.

In order to prove the security of our RIBE construction, we define a special security for IBE as follows:

- **Multi-Identity Selective Security**: For any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, there is a negligible function $\text{negl}(\cdot)$ such that the advantage of $\mathcal{A}$ satisfies:
A Generic Construction of Revocable Identity-Based Encryption

$$\text{Adv}^{\text{IND-msID-CPA}}_A = |\text{Pr}[\text{IND-msID-CPA}(A) = 1] - \frac{1}{2}| \leq \text{negl}(\lambda)$$

where IND-msID-CPA(A) is shown in Figure 2.

It is obvious that selective security is a special case of multi-identity selective security when there is only one challenge identity.

Experiment IND-sID-CPA(A):

1. \(id^* \leftarrow A_1(1^\lambda)\)
2. \((PP, MK) \leftarrow \text{Setup}(1^\lambda)\)
3. \((\mu_0, \mu_1) \leftarrow A_2^{\text{KeyGen}(MK, \cdot)}(PP)\) where \(|\mu_0| = |\mu_1|\) and for each query \(id\) by \(A_2\) to KeyGen(MK, ·) we have that \(id \neq id^*\).
4. \(\beta \leftarrow \{0, 1\}\)
5. \(c^* \leftarrow \text{Enc}(PP, id^*, \mu_\beta)\)
6. \(\beta' \leftarrow A_3^{\text{KeyGen}(MK, \cdot)}(PP, c^*)\) and for each query \(id\) by \(A_3\) to KeyGen(MK, ·) we have that \(id \neq id^*\).
7. Output 1 if \(\beta = \beta'\) and 0 otherwise.

Fig. 1. The selective security experiment of IBE

Selective Security Implies Multi-Identity Selective Security

**Lemma 1** If no PPT adversaries against the selective (adaptive) security then there exists no PPT adversaries can break the multi-identity selective (adaptive) security.

**Proof.** Since the proof for the selective-identity security and that for adaptive identity security are essentially the same, we only show the proof for the former.

We prove the lemma by hybrid argument. First, we define \(q+1\) hybrid games \(\mathcal{H}_0, ..., \mathcal{H}_q\) where \(\mathcal{H}_0\) is the real game and for all \(i \in [q]\), \(\mathcal{H}_i\) is the same as \(\mathcal{H}_{i-1}\) except the way that the challenger generates the challenge ciphertext. In \(\mathcal{H}_i\), the challenger computes the challenge ciphertext as \(\{c^*_j \leftarrow \text{Enc}(PP, id^*_j, 0)\}_{j \in \{1, ..., i\}}\) and \(\{c^*_j \leftarrow \text{Enc}(PP, id^*_j, \mu_\beta)\}_{j \in \{i+1, ..., q\}}\) where 0 is an all-zeros string with the same length of \(\mu_0\) and \(\beta\) is randomly chosen from \(\{0, 1\}\). Let \(S_i\) denote the event
Fig. 2. The multi-identity selective security experiment of IBE

that the output of IND-msID-CPA game is 1 in $H_i$. In $H_q$, the challenge ciphertext is encryption of zeros so $Pr[S_q] = \frac{1}{2}$. We will show that $|Pr[S_{i-1}] - Pr[S_i]| \leq \text{negl}(\lambda)$ for all $i \in [q]$ and finish the proof. We construct a PPT algorithm $B$ such that $|Pr[S_{i-1}] - Pr[S_i]|$ is equal to the probability that $B$ breaks the selective security of IBE. The detail of the algorithm $B$ is as follows:

1. $A$ outputs $q$ challenge identities $id^*_1, ... , id^*_q$, $B$ sends $id^*_i$ to its challenger
2. $B$’s challenger sends the public parameter $PP$ to $B$ and $B$ forwards it to $A$.
3. $A$ queries secret key for identity $id$, $B$ makes secret key query for $id$ and sends $sk_{id}$ to $A$. Note that $id \notin \{id^*_1, ... , id^*_q\}$. Then $A$ sends two plaintext $(\mu_0, \mu_1)$ with the same length.
4. $B$ randomly chooses a bit $\beta$ and sends $(0, \mu_\beta)$ to its challenger, where $|0| = |\mu_0| = |\mu_1|$. The challenger randomly chooses a bit $b$ and outputs $c^*_i = \text{Enc}(PP, id^*_i, 0)$ if $b = 0$ and $c^*_i = \text{Enc}(PP, id^*_i, \mu_\beta)$ if $b = 1$. Then, $B$ computes $\{c^*_j \leftarrow \text{Enc}(PP, id^*_j, 0)\}_{j \in \{1,...,i-1\}}$ and $\{c^*_j \leftarrow \text{Enc}(PP, id^*_j, \mu_\beta)\}_{j \in \{i+1,...,q\}}$. Finally, it outputs $c^* = (c^*_1, ... , c^*_q)$.
5. $B$ answers the secret key queries as Step 3. $A$ outputs a guess $\beta'$ of $\beta$. $B$ outputs $b' = 0$ if $\beta' = \beta$ and outputs $b' = 1$ otherwise.
6. Output 1 if $b' = b$. 

Experiment IND-msID-CPA($A$) :

1. $id^*_1, ... , id^*_q \leftarrow A_1(1^\lambda)$, where $q$ is a polynomial of $\lambda$.
2. $(PP, MK) \leftarrow \text{Setup}(1^\lambda)$
3. $(\mu_0, \mu_1) \leftarrow A_2^{\text{KeyGen}(MK, \cdot)}(PP)$ where $|\mu_0| = |\mu_1|$ and for each query $id$ by $A_2$ to $\text{KeyGen}(MK, \cdot)$ we have that $id \notin \{id^*_1, ... , id^*_q\}$
4. $\beta \leftarrow \{0, 1\}$
5. $\{c^*_i \leftarrow \text{Enc}(PP, id^*_i, \mu_\beta)\}_{i \in [q]}$
6. $\beta' \leftarrow A_3^{\text{KeyGen}(MK, \cdot)}(PP, c^*_1, ..., c^*_q)$ and for each query $id$ by $A_3$ to $\text{KeyGen}(MK, \cdot)$ we have that $id \notin \{id^*_1, ... , id^*_q\}$
7. Output 1 if $\beta = \beta'$ and 0 otherwise.
Note that if \( b = 0 \), \( B \) perfectly simulates the challenger in \( H_i \), and otherwise, it perfectly simulates that in \( H_{i-1} \). Moreover, the probability that \( b' = b \) satisfies:

\[
\Pr[b' = b] = \frac{1}{2} \Pr[b' = b | b = 0] + \frac{1}{2} \Pr[b' = b | b = 1]
\]

\[
= \frac{1}{2} \Pr[b' = b | b = 0] + \frac{1}{2} \Pr[b' = b | b = 1] + \frac{1}{2} (1 - \Pr[b' \neq b | b = 1])
\]

\[
= \frac{1}{2} + \frac{1}{2} \left( \Pr[\beta' = \beta | b = 0] - \Pr[\beta' = \beta | b = 1] \right)
\]

\[
= \frac{1}{2} + \frac{1}{2} \left( \Pr[S_i] - \Pr[S_{i-1}] \right)
\]

The selective security of IBE guarantees that \( |\Pr[b' = b] - \frac{1}{2}| \leq \text{negl}(\lambda) \) so that \( |\Pr[S_i] - \Pr[S_{i-1}]| \leq \text{negl}(\lambda) \) for all \( i \in \ell \). Hence, \( |\Pr[S_0] - \Pr[S_q]| = \frac{1}{2} \leq \text{negl}(\lambda) \). We complete the proof.

3 Generic Construction of Revocable Identity-Based Encryption

3.1 Definition and Security Model

A revocable IBE scheme has seven probabilistic polynomial-time (PPT) algorithms (Setup, KeyGen, KeyUpd, GenDk, Encrypt, Decrypt, Revoke) with associated message space \( \mathcal{M} \), identity space \( \mathcal{ID} \), and time space \( \mathcal{T} \).

- **Setup(1^{\lambda}, N)**: This algorithm takes as input a security parameter \( \lambda \) and a maximal number of users \( N \). It outputs a public parameter \( PP \), a master secret key \( MK \), a revocation list \( RL \) (initially empty), and a state \( st \).

- **KeyGen(PP, MK, id, st)**: This algorithm takes as input the public parameter \( PP \), the master secret key \( MK \), an identity \( id \), and the state \( st \). It outputs a secret key \( sk_{id} \) and an update state \( st \).

- **KeyUpd(PP, MK, t, RL, st)**: This algorithm takes as input the public parameter \( PP \), the master secret key \( MK \), a key update time \( t \in \mathcal{T} \), the revocation list \( RL \), and the state \( st \). It outputs a key update \( ku_t \).

- **GenDk(sk_{id}, ku_t)**: This algorithm takes as input a secret key \( sk_{id} \) and the key update \( ku_t \). It outputs a decryption \( dk_{id,t} \) or a special symbol \( \bot \) indicating that \( id \) was revoked.

- **Encrypt(PP, id, \mu)**: This algorithm takes as input the public parameter \( PP \), an identity \( id \), and a message \( \mu \in \mathcal{M} \). It outputs a ciphertext \( c \).
• **Decrypt**(PP, dk_{id,t}, c) : This algorithm takes as input the public parameter PP, a decryption secret key sk_{id,t} and a ciphertext. It outputs a message \( \mu \in \mathcal{M} \).

• **Revoke**(id, t, RL) : This algorithm takes as input an identity id, a revocation time \( t \in \mathcal{T} \) and the revocation list RL. It outputs a revocation list RL_t.

It satisfies the following conditions:

- **Correctness:** For all \( \lambda \) and polynomials (in \( \lambda \)) \( N \), all PP and MK output by setup algorithm Setup, all \( \mu \in \mathcal{M} \), id \( \in \mathcal{ID} \), t \( \in \mathcal{T} \) and all possible valid states st and revocation list RL, if identity id was not revoked before or, at time t then there exists a negligible function \( \text{negl}(\cdot) \) such that the following holds:

\[
\Pr[\text{Decrypt}(sk_{id,t}, \text{Encrypt}(PP, id, t, \mu)) = \mu] \geq 1 - \text{negl}(\lambda)
\]

where \((sk_{id}, st) \leftarrow \text{KeyGen}(PP, MK, id, st), ku_t \leftarrow \text{KeyUp}(PP, MK, t, RL, st)\) and \(dk_{id,t} \leftarrow \text{GenDk}(sk_{id}, ku_t)\).

- **Selective Security:** For any PPT adversary \( A = (A_1, A_2, A_3) \), there is a negligible function \( \text{negl}(\cdot) \) such that the advantage of \( A \) satisfies:

\[
\text{Adv}^{\text{IND-sRID-CPA}}_A = |\Pr[\text{IND-sRID-CPA}(A) = 1] - \frac{1}{2}| \leq \text{negl}(\lambda)
\]

where IND-sRID-CPA\( (A) \) is shown is Figure 3.

### 3.2 A Generic Construction from IBE

**Basic Intuition.** The key observation behind our construction is that BGK-RIBE utilized a tree-based approach which makes the scheme scalable. Recall that Path(id) denote the set of nodes on the path from id to root. KGC issues secret key for id the id-component decryption key for all nodes in Path(id). Moreover, there was a KUNode algorithm which outputs a minimal set S of nodes that contains an ancestor of all leaves corresponding to non-revoked users and the key update is the t-component decryption key for all nodes in S. In BGK-RIBE, only non-revoked users can derive decryption key sk_{id,t} by combining the id-component decryption key and the t-component decryption key for one ancestor of id. Inspired by the idea of tree-based approach, we use secret key extractions to generate key updates. Specifically, we divide our message \( \mu \)
Experiment IND-sRID-CPA($A$):

1. $(id^*, t^*) \leftarrow A_1(1^\lambda)$
2. $(PP, MK) \leftarrow \text{Setup}(1^\lambda)$
3. $(\mu_0, \mu_1) \leftarrow A^\text{KeyGen(MK, ·, KeyUp(PP, MK, ·, RL, st), Revoke(·))}_3(PP)$ where $|\mu_0| = |\mu_1|$
4. $\beta \leftarrow \{0, 1\}$
5. $c^* \leftarrow \text{Encrypt}(PP, id^*, t^*, \mu_\beta)$
6. $\beta' \leftarrow A^\text{KeyGen(MK, ·, KeyUp(PP, MK, ·, RL, st), Revoke(·))}_3(PP, c^*)$
7. Output 1 if $\beta = \beta'$ and 0 otherwise.

The following restriction must hold:

- $\text{KeyUp}(PP, MK, ·, RL, st)$ and $\text{Revoke}(·)$ can be queried on time which is greater than or equal to the time of all previous queries, i.e., the adversary is allowed to query only in non-decreasing order of time. Also, the oracle $\text{Revoke}(·)$ cannot be queried at time $t$ if $\text{KeyUp}(PP, MK, ·, RL, st)$ was queried on time $t$.
- If $\text{KeyGen}(MK, ·)$ was queried on identity $id^*$, then $\text{Revoke}(·)$ must be queried on time $t$ for some $t \leq t^*$, i.e. $(id^*, t)$ must be on revocation list $RL$ when $\text{KeyUp}(PP, MK, ·, RL, st)$ is queried on $t^*$.

Fig. 3. The selective security experiment of Revocable IBE
into \((\mu_0, \mu_1)\) where \(\mu_0\) and \(\mu_1\) are random with the condition \(\mu = \mu_0 + \mu_1\). So no information about \(\mu\) is revealed if only knowing \(\mu_0\) or \(\mu_1\). Our ciphertext can be divided into two parts, one part is the encryption of \(\mu_0\) under the receiver’s identity \(\text{id}\), the other part is encryption of \(\mu_1\) under identities \(t\|\theta\) for all \(\theta \in \text{Path(id)}\). So \(\mu_1\) can be recovered by any one of secret keys of \(\{sk_{t\|\theta}\}_{\theta \in \text{Path(id)}}\). Every user is issued a secret key \(sk_{\text{id}}\) as the long term secret key. To generate the key update for time \(t\), KGC extract secret keys for all identities \(t\|v\) where \(v\) is the node in \(\text{KUNode(t, RL, BT)}\). Hence, all users can obtain \(\mu_0\) by decrypting the first part of ciphertexts while only non-revoked users obtain \(\mu_1\) by decrypting the second part of ciphertexts using \(sk_{t\|\theta}\) in \(\text{ku}_t\) where \(\theta \in \text{Path(id)}\).

**Definition 1 (KUNode Algorithm [4])** This algorithm takes as input a binary tree \(BT\), revocation list \(RL\) and time \(t\), and outputs a set of nodes. Let \(\theta_{left}\) and \(\theta_{right}\) denote the left and right child of node \(\theta\), where \(\theta\) is a non-leaf node. The description of \(\text{KUNode}\) is as follows:

\[
\text{KUNode}(BT, RL, t):
\]

\[X, Y \leftarrow \emptyset\]
\[\forall (id_i, t_i) \in RL\]
\[\text{if } t_i \leq t \text{ then add } \text{Path}(id_i) \text{ to } X\]
\[\forall \theta \in X\]
\[\text{if } \theta_{left} \notin X \text{ then add } \theta_{left} \text{ to } Y\]
\[\text{if } \theta_{right} \notin X \text{ then add } \theta_{right} \text{ to } Y\]
\[\text{If } Y = \emptyset \text{ then add root to } Y\]
\[\text{Return } Y\]

Figure 4 gives a simple example to help the readers easily understand \(\text{KUNode}(BT, RL, t)\).

In the example, identities 001 and 100 are revoked. \(X = \text{Path}(001) \cup \text{Path}(100) = \{\text{root}, 00, 001, 1, 10, 100\}\), and \(Y = \{01, 11, 000, 101\}\). Intuitively, for all non-revoked identities \(id\) such that \(\text{Path}(id) \cap Y \neq \emptyset\) while for revoked identities such that \(\text{Path}(001) \cap Y = \emptyset\) and \(\text{Path}(100) \cap Y = \emptyset\).

**Detailed Construction.** Let \((\text{IBE.Setup}, \text{IBE.Enc}, \text{IBE.KeyGen}, \text{IBE.Dec})\) be an IBE scheme that supports \(\mathcal{ID} = \{0, 1\}^{\ell, 2\ell}\). There is a generic method to extend any IBE supporting identity space \(\mathcal{ID}'\) to handle arbitrary identities \(id \in \{0, 1\}^*\) by first hashing \(id\) using a collision resistant hash function \(H : \{0, 1\}^* \rightarrow \mathcal{ID}'\) prior to key generation and encryption [5]. Hence, the IBE
scheme supporting identity space \( ID' \) with a collision resistant hash function \( H : \{0, 1\}^* \rightarrow ID' \) can be applied for our construction. We assume IBE scheme has the plaintext space \( M \) which is finite and forms an abelian group with the group operation “+”.

Utilizing the above IBE scheme, we will show how to construct a RIBE scheme \( \Pi = (\text{Setup}, \text{Encrypt}, \text{Decrypt}, \text{KeyGen}, \text{KeyUp}, \text{GenDk}, \text{Revoke}) \) as follows. In our RIBE scheme, the plaintext space is the same with the underlying IBE scheme and identity space is \( \{0, 1\}^\ell \). Moreover, we assume the time period space \( T \) is a subset of the identity space, i.e. \( T \subseteq \{0, 1\}^\ell \).

- **Setup** \((1^\lambda) \rightarrow (PP, MK)\): This algorithm takes the security parameter \( 1^\lambda \) as input and runs \((\text{IBE}.PP, \text{IBE}.MK) \leftarrow \text{IBE}.\text{Setup}(1^\lambda)\). It sets the public parameter \( PP = \text{IBE}.PP \), master secret key \( MK = \text{IBE}.MK \) and secret state \( st = \text{IBE}.MK \). The following algorithms implicitly take \( PP \) as input.
- **Encrypt** \((PP, id, t, \mu) \rightarrow c\): Randomly sample a pair of plaintexts \((\mu_0, \mu_1) \in M^2\) with the condition that \( \mu = \mu_0 + \mu_1 \). Then it computes \( c_0 = \text{IBE}.\text{Enc}(PP, id, \mu_0) \) and \( \{c_i = \text{IBE}.\text{Enc}(PP, t[i\|id[1..i]], \mu_1)\}_{i \in [\ell]} \). Finally, it outputs the ciphertext \( c = (c_0, ..., c_\ell) \).
- **KeyGen** \((MK, id) \rightarrow sk_{id}\): It runs \( sk_{id} \leftarrow \text{IBE}.\text{KeyGen}(MK, id) \).
- **KeyUp** \((t, RL_t, st) \rightarrow ku_t\): Let BT be a complete binary tree of depth \( \ell \). Every identity id in the identity space \( \{0, 1\}^\ell \) can be viewed as a leaf node of BT. For each node \( \theta \in \text{KUNode}(BT, RL, t) \), compute \( sk_{i\|\theta} \leftarrow \text{IBE}.\text{KeyGen}(\text{IBE}.MK, t[i\|\theta]) \). It outputs \( ku_t = \{(\theta, sk_{i\|\theta})\}_{\theta \in \text{KUNode}(BT, RL, t)} \).
- **GenDk** \((sk_{id}, ku_t) \rightarrow sk_{id,t}\): Parse \( ku_t \) as \( \{(\theta, sk_{i\|\theta})\}_{\theta \in \text{KUNode}(BT, RL, t)} \). If no node \( \theta \in \text{Path}(id) \), return \( \bot \). Otherwise, pick the node \( \theta \in \text{Path}(id) \) and output \( sk_{id,t} = (i, sk_{id}, sk_{i\|\theta}) \) where \( i = |\theta| \) is the length of \( \theta \).
• Decrypt($c, sk_{id,t}) \rightarrow \mu$ : Parse $c$ as $(c_0, ..., c_f)$ and $sk_{id,t}$ as $(i, sk_{id}, sk_{t||\theta})$. Then, compute $\mu_0 \leftarrow \text{IBE}.\text{Dec}(sk_{id}, c_0)$ and $\mu_1 \leftarrow \text{IBE}.\text{Dec}(sk_{t||\theta}, c_i)$. Finally, output $\mu = \mu_0 + \mu_1$.

• Revoke($t, RL, id$) $\rightarrow (RL_t)$ : Add the pair $(id, t)$ to the revocation list by $RL_t \leftarrow RL \cup \{(id, t)\}$ and output $RL_t$.

3.3 Correctness

The correctness of the RIBE construction is guaranteed by the correctness of the underlying IBE.

3.4 Security Analysis

**Theorem 1** The revocable IBE is selectively (adaptively) secure if the underlying IBE scheme is selectively (adaptively) secure.

**Proof.** We will prove the selective-identity security and the proof for adaptive-identity security are exactly the same. For any PPT adversary against the selective security of revocable IBE, we can construct a PPT algorithm $B$ against the selective security of the underlying IBE scheme. $B$ randomly guesses an adversarial type among the following two types which are mutually exclusive and cover all possibilities:

1. Type-1 adversary: $A$ issues a secret key query for $id^*$ hence $id^*$ has been revoked before $t^*$.
2. Type-2 adversary: $A$ does not issue a secret key query for $id^*$.

Note that $B$’s guess is independent of the attack that $A$ chooses, so the probability that $B$ guesses right is $\frac{1}{2}$. We separately describe $B$’s strategy by its guess.

**Type-1 adversary:** We will show that if adversary $A_1$ makes a Type-1 attack successfully, there exists an adversary $B_1$ breaking the multi-identity selective security of IBE defined in definition 2. $B_1$ proceeds as follows:

• Setup: The adversary first commits an identity $id^*$ and a time period $t^*$ to $B_1$. Upon receiving the identity $id^*$ and time period $t^*$ committed by $A_1$, $B_1$ commits identities $\{t^*||id^*_{[1, i]}\}_{i \in [f]}$ to its challenger. $B_1$ then obtains a public parameter $PP$ from its challenger and sends it to $A_1$. 
KeyGen: When receiving a secret key query for id, B1 queries secret key extraction oracle for id. Since \(|id| = \ell\) and \(|t^*|/|id| \geq \ell + 1\) for all \(i \in [\ell]\), id \(\notin \{t^*|/|id| \}_i \in [\ell]\).

Revoke: B1 receives (id,t) from A1, and adds (id,t) to RL.

KeyUp: Upon receiving t, if t = t* and (id*,t) \(\notin\) RL+, then abort. Otherwise, B1 makes secret key queries for identities \({\ell}|/|\theta|/|\theta \in KUNode(BT,RL+)\) and sends \({\theta, sk_{\theta(i)}}\) \(\in KUNode(BT,RL+)\) to A1. Note that id* has been revoked before t*, which means id’ \(\notin\) KUNode(BT,RL+,t*) for all \(i \in [\ell]\), so that B1 never queries secret keys for identities \({t^*|/|id| \}_i \in [\ell]\) committed to its challenger.

Challenge: A1 outputs two plaintexts \(\mu_0\) and \(\mu_1\) with the same length. B1 randomly samples \(\mu \leftarrow \mathcal{M}\) and sends \(\mu_0 = \mu_0 - \mu\) and \(\mu_1' = \mu_1 - \mu\) as the challenge plaintexts. The challenger randomly chooses a challenge bit \(\beta\) and sends the challenge ciphertexts \({c^*_\beta = IBE.Enc(PP,t^*,/\theta|/|id| \}_i \in [\ell]}\) to B1. B1 then computes \(c^*_\beta = IBE.Enc(PP,\text{id}^*,\mu)\) and sends \(c^*_\beta = (c^*_0, ..., c^*_\ell)\) to A1.

Guess: A1 outputs a guess bit \(\beta'\) and B1 set \(\beta'\) as its guess.

Note that B1 perfectly simulates A1’s view so that B1’s challenge bit is also A1’s challenge bit. B1 just forwards A1’s guess so the probability that B1 wins in IND-msID-CPA is equal to the probability that A1 wins in IND-sRID-CPA. Due to Lemma 1, the probability that A1 wins in IND-sRID-CPA is negligible since the underlying IBE is selectively secure.

Type-2 adversary: If there exists an adversary A2 who makes a Type-2 attack successfully, we can construct an adversary B2 breaking selective security of the underlying IBE. B2 proceeds as follows:

- Setup: Upon receiving the identity \(\text{id}^*\) and time period \(t^*\) committed by A2, B2 commits identity \(\text{id}^*\) to its challenger. B2 then obtains a public parameter PP from its challenger and sends it to A2.

- KeyGen: When receiving a secret key query for id, B2 just forwards the secret key query to its challenger and sends the challenger’s response to A2. Note that A2 never make a secret key query for \(\text{id}^*\).

- Revoke: B2 receives (id,t) from A2, and adds (id,t) to RL.

- KeyUp: When A2 makes a key update query for time t, B2 makes secret key queries for all identities \({t}|/|\theta|/|\theta \in KUNode(BT,RL+)\) and sends the response \({\theta, sk_{\theta(i)}}\) \(\in KUNode(BT,RL+)\) to A2.

- Challenge: A2 outputs two plaintexts \(\mu_0\) and \(\mu_1\) with the same length. B1 randomly samples \(\mu \leftarrow \mathcal{M}\) and sends \(\mu_0' = \mu_0 - \mu\) and \(\mu_1' = \mu_1 - \mu\) as the chal-
lenge plaintexts. $B_1$ receives the challenge ciphertext $c_0^* = \text{IBE.E}nc(PP, id^*, \mu'_\beta)$ where $\beta$ is $B_2$’s challenge bit chosen randomly by its challenger. $B_2$ then computes $\{c_i^* = \text{IBE.E}nc(PP, t^* || id^*_{i, i}, \mu)\}_{i \in [\ell]}$ and sends $c^* = (c_0^*, ..., c_\ell^*)$ to $A_2$.

- **Guess:** $A_2$ outputs a guess bit $\beta'$ and $B_2$ sets $\beta'$ as its guess. Note that $B_2$ perfectly simulates $A_2$’s view so that $B_2$’s challenge bit is also $A_2$’s challenge bit. $B_2$ just forwards $A_2$’s guess so the probability that $B_2$ wins in IND-sID-CPA game is equal to the probability that $A_2$ wins in IND-sRID-CPA game.

When we put the results for two types of adversary together, we can conclude that the revocable IBE is selectively secure if the underlying IBE is selectively secure.

## 4 Discussion

**Server-Aided.** In RIBE schemes, non-revoked user should receive the key update in every time period. Fortunately, our scheme is server-aided so that almost all the workload on users is taken over by a untrusted server who should perform correct operations and give correct results to the users. In our scheme, given the key update $k_u = \{(\theta, sk_t, \theta)\}$ where $\theta \in \text{KUNode} (BT, RL_t, t)$ and a ciphertext $c = (c_0, ..., c_\ell)$ under identity $id$ and time $t$, the server chooses $\theta \in \text{Path(id)}$ and computes $\mu' \leftarrow \text{Dec}(sk_t || \theta, c_i)$ where $i = |\theta|$. Finally, the server sends $(c_0, \mu')$ to the receiver.

**Short Ciphertext.** The size of ciphertext is logarithmic in the number of users in our construction. Fortunately, we can replace the underlying IBE scheme with IBBE scheme and there exists IBBE schemes with constant size of ciphertext and secret key. The intuition of security proof is that the selective (adaptive) security of IBBE implies multi-identity selective (adaptive) security of IBE.

**RIBE with DKER.** It is obvious that our construction is not decryption key exposure resistance. Recently, Katsumata et al. [25] presented a generic construction of RIBE with DKER from any RIBE without DKER. Therefore, we can obtain a RIBE with DKER from any IBE by applying this generic conversion in [25].
5 Conclusion

In this paper, we proposed a generic conversion from IBE to RIBE without DKER. Applying the conversion in [25], we obtained a generic conversion from IBE to RIBE without DKER. Our RIBE construction inherits the security of the underlying IBE scheme, therefore, our construction implies the first RIBE from quadratic residues modulo composite and the first adaptively secure RIBE from lattices. Furthermore, our conversion is efficient and flexible. The sizes of public parameters and secret keys are the same as those of the underlying IBE scheme. In the server-aided model, the communication and computation overheads are the same as those of the underlying IBE scheme in the server-aided model. There is a tradeoff between the size of public parameters and the size of ciphertexts if we replace the underlying IBE with appropriate IBBE.

References


