Generic Constructions of RIBE via Subset Difference Method

Xuecheng Ma[1,2] and Dongdai Lin[1,2]

[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,ddlin}@iie.ac.cn

Abstract. Revocable identity-based encryption (RIBE) is an extension of IBE which can support a key revocation mechanism, and it is important when deploying an IBE system in practice. Boneh and Franklin (Crypto'01) presented the first generic construction of RIBE, however, their scheme is not scalable where the size of key updates is linear in the number of users in the system. The first generic construction of RIBE is presented by Ma and Lin with complete subtree (CS) method by combining IBE and hierarchical IBE (HIBE) schemes. Recently, Lee proposed a new generic construction using the subset difference (SD) method by combining IBE, identity-based revocation (IBR), and two-level HIBE schemes.
In this paper, we present a new primitive called Identity-Based Encryption with Ciphertext Delegation (CIBE) and propose a generic construction of RIBE scheme via subset difference method using CIBE and HIBE as building blocks. CIBE is a special type of Wildcarded IBE (WIBE) and Identity-Based Broadcast Encryption (IBBE). Furthermore, we show that CIBE can be constructed from IBE in a black-box way. Instantiating the underlying building blocks with different concrete schemes, we can obtain a RIBE scheme with constant-size public parameter, ciphertext, private key and $O(r)$ key updates in the selective-ID model. Additionally, our generic RIBE scheme can be easily converted to a sever-aided RIBE scheme which is more suitable for lightweight devices.

Key words: Generic Construction, Revocable IBE, Subset Difference, DKER

1 Introduction

Identity-Based Encryption (IBE) was introduced by Shamir [47], 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 [9] in the random oracle model [4]. Since then, there are many follow-up works [6, 7, 50, 21, 51, 15, 10, 2, 3, 11–13, 22, 52, 53, 19]. A hierarchical IBE (HIBE) scheme [23, 25] generalizes the concept of IBE by forming levels of a hierarchy. For an ℓ-level HIBE, a hierarchical identity is a vector of maximal ℓ identities, and a user at level i can generate a secret key for its descendants at level j (where $i < j \leq \ell$).

To address the challenge of key revocation in IBE setting, Boneh and Franklin [9] presented the notion of revocable IBE and proposed a naive method to add 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., $\mathsf{id}||\mathtt{T}$ and the Key Generation Center (KGC) issues the private key $\mathsf{sk}_{\mathsf{id}||\mathtt{t}}$ for each non-revoked user 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 does not scale well when there are a large number of users.

Boldyreva, Goyal and Kumar [5] proposed the first scalable revocable IBE (RIBE) scheme by combining the fuzzy IBE scheme of Sahai and Waters [43] with the complete subtree (CS) method [36]. In the definition of security in BGK-RIBE, the adversary is only given access to the secret key oracle, the revocation oracle and the key update oracle. Seo and Emura [44, 46] introduced a security notion called decryption key exposure resistance (DKER) which captures the realistic attack that decryption keys may be leaked. 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 which are encrypted for different time periods. It attracted many follow-up works concerning R(H)IBE schemes with DKER [20, 27, 29, 31, 32, 35, 39, 40, 42, 46, 49].

Server-aided RIBE [41, 16, 37] 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 needs to store a short long-term private key without having to communicate with KGC.

Ma and Lin [34] proposed a generic construction of RIBE using complete subtree method by combining IBE and HIBE in a black-box way which solved the open problem presented in [44]. In their first scheme, an update key consists of $O(r \cdot \ell)$ IBE private keys and a ciphertext consists of $O(\ell)$ IBE ciphertexts where r is the number of revoked users and n is the bit length of an identity. And they also made some optimization using HIBE or IBBE [17] which makes the ciphertext size constant. Currently, Lee [30] proposed a generic RIBE scheme with the subset difference method by using IBE, identity-based revocation (IBR), and two-level HIBE schemes as basic building blocks. Their scheme reduced the size of an update key from $O(r \cdot \ell)$ key elements to $O(r)$ key elements but the ciphertext size increased to $O(\ell^2)$ number of IBE and IBR ciphertexts. In addition, they showed how to reduce the ciphertext size by extending their generic RIBE scheme to use the more efficient LSD method instead of using the SD method.

1.1 Our Contributions.

In order to construct a generic RIBE scheme using SD method, we first present a new primitive called identity based encryption with ciphertext delegation (CIBE). Contrary to HIBE where an identity secret key can decrypt ciphertexts encrypted under its *descendants*, an identity secret key can decrypt ciphertexts encrypted under its *ancestors* in a CIBE scheme. In addition, the plaintext encrypted under an identity id is confidential if the adversary does not know the secret key of id or descendants of id. It is obvious that the new primitive CIBE is a special type of wildcarded IBE (WIBE)[1] where the wildcard "*" just appears at the end portion of the pattern. It can also be viewed as a special type of IBBE where we encrypt a plaintext under all descendants of id[3]. Moreover, we will show that CIBE can be constructed from IBE in a black-box way.

In this paper, we propose a generic construction of RIBE with SD method by combining CIBE and a two-level HIBE. Our technique and building blocks are totally different from Lee's generic RIBE scheme. In our generic construction, the key update size is $O(r)$ CIBE keys and the ciphertext is $O(\ell^2)$ CIBE ciphertexts and one HIBE ciphertext. Furthermore, CIBE can be constructed from IBE in a black-box way so we can give a generic construction of RIBE with SD method by using IBE and HIBE as building blocks. However the secret key size of the generic CIBE scheme from IBE is $O(\ell)$ which result in a $O(r\ell)$ size of key update in the generic RIBE scheme. Although the key update is not short as that of Lee's construction, it shows the possibility that generically construct RIBE with SD method using IBE and HIBE and we wish to give a CIBE scheme with shorter secret key from IBE in the future. Furthermore, we can construct CIBE from WIBE (IBBE), our generic RIBE scheme consists of $O(r)$ WIBE (IBBE) secret keys in key update and $O(\ell^2)$ WIBE (IBBE) ciphertexts and one HIBE ciphertext in a ciphertext. In addition, the layered SD (LSD) method can be applied to a generic RIBE scheme which reduces the ciphertext and the update key of our generic RIBE scheme to $O(\ell^{1.5})$ CIBE ciphertexts and $4r$ CIBE private keys, respectively. Last but not least, we can reduce the ciphertext size by using IBBE solves the open problem presented by Lee [30]. Nota that it is difficult to reduce the ciphertext size in Lee's scheme since it uses an IBR scheme. Instantiating the underlying IBBE and HIBE schemes with proper concrete schemes, we can obtain a RIBE scheme with constant-size public parameter, ciphertext, private key and $O(r)$ key updates in the selective-ID model.

1.2 Our Technique

Let us first describe the generic RIBE scheme using CS method proposed by Ma and Lin. Ma and Lin observed that the design principle of CS method in the symmetric setting [36]. In symmetric Broadcast encryption [36], every user corresponds to a leaf node in a complete binary tree and holds all secret keys corresponding to the nodes in the path from root to the associated leaf. The non-revoked users are covered by complete subtrees and the plaintext is encrypted by secret keys

[3] In fact, WIBE is a special type of IBBE.

corresponding the root node of the subtrees covers all the non-revoked users. A user is not revoked if and only if there is a ciphertext encrypting the plaintext under a secret key corresponding a node in path from the root to the associated leaf node. In RIBE setting, Ma and Lin view an identity id as a leaf in a complete binary tree with depth $|\mathsf{id}|$. A plaintext is encrypted under the identifiers of nodes in the path from the root to id using IBE and KGC broadcast a set of IBE secret keys associated with the root node of the complete subtrees which cover all non-revoked users. The key update consists of secret key of one ancestor of id iff id is not revoked. We only describe the behind idea for realizing revocation mechanism the scheme of Ma and Lin. For the security reason, they divided the plaintexts into two secret shares, one is encrypted using HIBE and the other is encrypted using IBE under all ancestors of id.

Unlike CS method that covers non-revoked users by complete subtrees, SD method covers non-revoked users using subsets $CV_R = \{S_{i,j}\}$ where $S_{i,j}$ is presented by two nodes (v_i,v_j) and v_i is an ancestor of v_j and $S_{i,j}$ contains all leaves which are descendants of v_i but not of v_j in every subtree $\mathcal{T}_i$ where $\mathcal{T}_i$ is a complete subtree rooted at the ancestor of v_{id} in depth i. The Assign algorithm assigns secret keys corresponding nodes which are adjacent to v_{id} but not ancestors of v_{id}. Let $PV_{\mathsf{id}} = \{S'_{i,j}\}$ denote the node subset assigned to id where $S'_{i,j}$ is presented by (v'_i,v'_j). The assign algorithm and cover algorithm guarantees that id is not revoked iff there exists $S_{i,j} \in CV_R$ and $S'_{i,j} \in PV_{\mathsf{id}}$ such that $v_i = v'_i$ and v'_j is an ancestor of v_j. In order to apply SD method to RIBE, we encrypt the plaintext under corresponding nodes in PV_{id} using CIBE and KGC broadcasts CIBE secret keys $\{\mathsf{sk}_{S_{i,j}}\}_{S_{i,j} \in CV_R}$. The ciphertext delegation property of CIBE guarantees that the ciphertext can be decrypted iff there is an identity in the ciphertext which is a prefix of $S_{i,j}$.

1.3 Related Works

Revocable IBE. The first revocable IBE scheme from any IBE was presented by Boneh and Franklin [9], however their proposal was not scalable. Boldyreva et al. [5] proposed the first scalable RIBE combining fuzzy IBE and CS method. A number of secure and efficient RIBE schemes using a broadcast method for key updates have been proposed [14, 28, 33, 40, 44, 48]. Most of the RIBE schemes follow the CS method for update keys, but Lee et al. [31] showed that an RIBE scheme with the SD method can be designed to reduces the size of update keys. Recently, Ma and Lin [34] proposed a generic RIBE construction with the CS method by combining IBE and HIBE schemes. Subsequently, Lee proposed a generic RIBE scheme with SD method using IBE, IBR and HIBE as building blocks.
Revocable HIBE. Seo and Emura [34] presented the first revocable HIBE (RHIBE) scheme with history-preserving updates, wherein a low-level user must know the history of key updates performed by ancestors in the current time period which makes the scheme very complex. Subsequently, Seo and Emura [45] presented a new method to construct RHIBE that implements history-free updates. After that, there are some follow-up works concerning about efficiency [32], stronger security [29] or assumptions without pairing [28].

2 Preliminaries

2.1 Notations

Throughout the paper we use the following notation: We use λ as the security parameter and write $\mathsf{negl}(\lambda)$ to denote that some function $f(\cdot)$ 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 $\mathsf{poly}(\lambda)$. 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 $[n]$ denotes $\{1, ..., n\}$. Let $\{0,1\}^{[i,j]}$ denotes all binary strings with length in $[i, j]$. For a bit string $a = (a_1, ..., a_n) \in \{0,1\}^n$, and $i, j \in [n]$ with $i \leq j$, we write $a_{[i,j]}$ to denote the substring $(a_i, ..., a_j)$ of a. For any two strings u and v, $|u|$ denote the length of u and $u||v$ denotes their concatenation. Let $\mathcal{BT}$ be a complete binary tree. For two strings s and t of length ℓ, we use $s =_* t$ to denote s matches t and $s \neq_* t$ to denote s does not match t. We define $s =_* t$ iff $s_i = t_i \vee t_i = *$ for all $i \in \{1, ..., \ell\}$ and $s \neq_* t$ iff $s_i \neq t_i \wedge t_i \neq *$ for some $i \in \{1, ..., \ell\}$.

2.2 Identity-Based Encryption with Ciphertext Delegation

An IBE with ciphertext delegation scheme (CIBE) consists of four algorithms Setup, KeyGen, Enc, and Dec, which are defined as follows:

1. $\mathsf{Setup}(1^\lambda)$:The setup algorithm takes as input a security parameter 1^λ and outputs a master key MK and public parameter PP.

- $\mathsf{KeyGen}(\mathsf{MK},\mathsf{id})$: This algorithm takes as input the master secret key MK and an identity $\mathsf{id} \in \{0,1\}^\ell$, it outputs the identity secret key $\mathsf{sk}_{\mathsf{id}}$.
- $\mathsf{Enc}(\mathsf{PP},\mathsf{id},\mu)$: This algorithm takes as input the public parameter PP, an identity $\mathsf{id} \in \{0,1\}^{[\ell_0,\ell_1]}$ where $\ell_0 < \ell_1$, and a plaintext μ, it outputs a ciphertext c.
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}},\mathsf{c})$: This algorithm takes as input a secret key $\mathsf{sk}_{\mathsf{id}}$ for identity id and a ciphertext c, it outputs a plaintext μ.

Correctness: The correctness of CIBE is defined as follows: For all security parameters 1^λ, two identities $\mathsf{id} \in \{0,1\}^{\ell_0'}$, $\mathsf{id}' \in \{0,1\}^{\ell_1'}$ and plaintext μ, the following holds:

$$\Pr[\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}'},\mathsf{Enc}(\mathsf{PP},\mathsf{id},\mu)) = \mu] = 1$$

where id is a prefix of id′, $(\mathsf{PP},\mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{MK},\mathsf{id})$.

Multi-Identity Adaptive Security: For any PPT adversary $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\cdot)$ such that the following holds:

$$Adv_{\mathcal{A}}^{\mathsf{IND\text{-}mCID\text{-}CPA}} = |\Pr[\mathsf{IND\text{-}mCID\text{-}CPA}(\mathcal{A}) = 1] - \tfrac{1}{2}| \leq \mathsf{negl}(\lambda)$$

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

If $q = 1$, we call the above experiment as single-identity adaptive security (IND-CID-CPA). Additionally, we can define the selective security analogously where the adversary first commit the challenge identities before obtaining the public parameter. Obliviously, single-identity security is a special case of multi-identity security. For the other direction, we will show that single-identity security implies multi-identity security.

Experiment $\mathsf{IND\text{-}mCID\text{-}CPA}(\mathcal{A})$:

1. $(\mathsf{PP},\mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mu_0,\mu_1,\mathsf{id}_1^*,...,\mathsf{id}_q^*) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MK},\cdot)}(\mathsf{PP})$ where q is a polynomial of λ, $|\mu_0| = |\mu_1|$ and for each query id to $\mathsf{KeyGen}(\mathsf{MK},\cdot)$ we have that any $\mathsf{id}' \in \{\mathsf{id}_1^*,...,\mathsf{id}_q^*\}$ is not a prefix of id (or equal to id).
3. $\beta \leftarrow \{0,1\}$
4. $\{\mathsf{c}_i^* \leftarrow \mathsf{Enc}(\mathsf{PP},\mathsf{id}_i^*,\mu_\beta)\}_{i\in[q]}$
5. $\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MK},\cdot)}(\mathsf{PP},\mathsf{c}_1^*,...,\mathsf{c}_q^*)$ and for each query id to $\mathsf{KeyGen}(\mathsf{MK},\cdot)$ we have that any $\mathsf{id}' \in \{\mathsf{id}_1^*,...,\mathsf{id}_q^*\}$ is not a prefix of id (or equal to id).
6. Output 1 if $\beta = \beta'$ and 0 otherwise.

Fig. 1. The multi-identity adaptive security experiment of CIBE

Lemma 1. *An CIBE scheme is multi-identity adaptively (selectively) secure if it is single-identity adaptively (selectively) secure.*

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

We prove the lemma by hybrid arguments. First, we define $q+1$ hybrid games $\mathcal{H}_0,...,\mathcal{H}_q$ where $\mathcal{H}_0$ is the real IND-mID-CPA 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 $\{\mathsf{c}_j^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}_j^*, 0)\}_{j\in\{1,...,i\}}$ and $\{\mathsf{c}_j^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}_j^*, \mu_\beta)\}_{j\in\{i+1,...,q\}}$ where $\mathsf{0}$ is an all-zeros string with the same length of μ_0 and β is randomly chosen from $\{0, 1\}$. Let S_i denote the event that the output of IND-mCID-CPA game is 1 in $\mathcal{H}_i$. In $\mathcal{H}_q$, the challenge ciphertext is encryption of zeros so $\Pr[S_q] = \frac{1}{2}$. We will show that $|\Pr[S_{i-1}] - \Pr[S_i]| \leq \mathsf{negl}(\lambda)$ for all $i \in [q]$ and finish the proof. We construct a PPT algorithm $\mathcal{B}$ such that $|\Pr[S_{i-1}] - \Pr[S_i]|$ is equal to the probability that $\mathcal{B}$ breaks single-identity adaptive-ID security of CIBE. The detail of the algorithm $\mathcal{B}$ is as follows:

1. $\mathcal{B}$'s challenger sends the public parameter PP to $\mathcal{B}$ and $\mathcal{B}$ forwards it to $\mathcal{A}$.
2. When $\mathcal{A}$ queries secret key for identity id, $\mathcal{B}$ makes secret key query for id and sends $\mathsf{sk}_{\mathsf{id}}$ to $\mathcal{A}$. Then $\mathcal{A}$ sends q challenge identities $\mathsf{id}_1^*, ..., \mathsf{id}_q^*$ and two plaintexts (μ_0, μ_1) with the same length.
3. $\mathcal{B}$ randomly chooses a bit β and sends $(\mathsf{0}, \mu_\beta, \mathsf{id}_i^*)$ to its challenger, where $|\mathsf{0}| = |\mu_0| = |\mu_1|$. The challenger randomly chooses a bit b and outputs $\mathsf{c}_i^* = \mathsf{Enc}(\mathsf{PP}, \mathsf{id}_i^*, \mathsf{0})$ if $b = 0$ and $\mathsf{c}_i^* = \mathsf{Enc}(\mathsf{PP}, \mathsf{id}_i^*, \mu_\beta)$ if $b = 1$. Then, $\mathcal{B}$ computes $\{\mathsf{c}_j^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}_j^*, \mathsf{0})\}_{j\in\{1,...,i-1\}}$ and $\{\mathsf{c}_j^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}_j^*, \mu_\beta)\}_{j\in\{i+1,...,q\}}$. Finally, it outputs $\mathsf{c}^* = (\mathsf{c}_1^*, ..., \mathsf{c}_q^*)$.
4. $\mathcal{B}$ answers the secret key queries as Step 2. $\mathcal{A}$ outputs a guess β' of β. $\mathcal{B}$ outputs $b' = 0$ if $\beta' = \beta$ and outputs $b' = 1$ otherwise.

Note that the identity id $\mathcal{A}$ submits to secret key oracle with the restriction that no one identity in $\{\mathsf{id}_1^*, ..., \mathsf{id}_q^*\}$ is a prefix of id. For all $i \in \{0, 1, ..., q\}$, $\mathcal{B}$ does not query secret key for id where there exists a challenge identity that is a prefix of id in $\mathcal{H}_i$. If $b = 0$, $\mathcal{B}$ perfectly simulates the challenger in $\mathcal{H}_i$, and otherwise, it perfectly simulates that in $\mathcal{H}_{i-1}$. Moreover, the probability that $b' = b$ satisfies:

$$\begin{aligned}
\Pr[b' = b] &= \Pr[b' = b|b = 0]\Pr[b = 0] + \Pr[b' = b|b = 1]\Pr[b = 1] \\
&= \frac{1}{2}\Pr[b' = b|b = 0] + \frac{1}{2}\Pr[b' = b|b = 1] \\
&= \frac{1}{2}\Pr[b' = b|b = 0] + \frac{1}{2}(1 - \Pr[b' \neq b|b = 1]) \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[\beta' = \beta|b = 0] - \Pr[\beta' = \beta|b = 1]) \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[S_i] - \Pr[S_{i-1}])
\end{aligned}$$

The single-identity adaptive security of CIBE guarantees that $|\Pr[b' = b] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$ so $|\Pr[S_i] - \Pr[S_{i-1}]| \leq \mathsf{negl}(\lambda)$ for all $i \in [q]$. Hence, $|\Pr[S_0] - \Pr[S_q]| = |\Pr[S_0] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$. We complete the proof.

We can construct CIBE from IBE in a black-box way. The Setup, Enc and Dec algorithms are the same of those of underlying IBE scheme. To generate a secret key for an identity id in CIBE scheme, we generate secret keys for all prefixes of id using KeyGen algorithm of IBE.

2.3 Wildcarded Identity-Based Encryption

A wildcarded identity-based encryption scheme consists of four probabilistic polynomial-time (PPT) algorithms (Setup, KeyGen, Enc, Dec) defined as follows:

- Setup(1^λ): This algorithm takes as input the security parameter 1^λ, and outputs a public parameter PP and a master secret key MK.
- KeyGen(MK,id): This algorithm takes as input the master secret key MK and an identity $\mathsf{id} \in \{0, 1\}^\ell$, it outputs the identity secret key $\mathsf{sk}_{\mathsf{id}}$.
- Enc(PP,P,μ): This algorithm takes as input the public parameter PP, a pattern $P \in \{0, 1, *\}^\ell$, and a plaintext μ, it outputs a ciphertext c.
- Dec($\mathsf{sk}_{\mathsf{id}}$, c): This algorithm takes as input a secret key $\mathsf{sk}_{\mathsf{id}}$ for identity id and a ciphertext c, it outputs a plaintext μ.

The following correctness and security properties must be satisfied:

– **Correctness:** For all security parameters 1^λ, any identity $\mathsf{id} \in \{0,1\}^\ell$, any pattern $P \in \{0,1,*\} * \ell$ and plaintext $\mu \in \mathcal{M}$, the following holds:

$$\Pr[\mathsf{Dec}(\mathsf{sk_{id}}, \mathsf{Enc}(\mathsf{PP}, P, \mu)) = \mu] = 1$$

where $\mathsf{id} =_* P$, $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{sk_{id}} \leftarrow \mathsf{KeyGen}(\mathsf{MK}, \mathsf{id})$.

– **Adaptive Security:** For any PPT adversary $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\cdot)$ such that the following holds:

$$Adv_{\mathcal{A}}^{\mathsf{IND\text{-}WID\text{-}CPA}} = |\Pr[\mathsf{IND\text{-}WID\text{-}CPA}(\mathcal{A}) = 1] - \tfrac{1}{2}| \leq \mathsf{negl}(\lambda)$$

where $\mathsf{IND\text{-}WID\text{-}CPA}(\mathcal{A})$ is shown in Figure 2.

Experiment $\mathsf{IND\text{-}WID\text{-}CPA}(\mathcal{A})$:

1. $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mu_0, \mu_1, P^*) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MK},\cdot)}(\mathsf{PP})$ where $|\mu_0| = |\mu_1|$ and for each query id to $\mathsf{KeyGen}(\mathsf{MK},\cdot)$ we have that $\mathsf{id} \neq_* P^*$.
3. $\beta \leftarrow \{0,1\}$
4. $\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathsf{PP}, P^*, \mu_\beta)$
5. $\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MK},\cdot)}(\mathsf{PP}, \mathsf{c}^*)$ and for each query id to $\mathsf{KeyGen}(\mathsf{MK},\cdot)$ we have that $\mathsf{id} \neq_* P^*$.
6. Output 1 if $\beta = \beta'$ and 0 otherwise.

Fig. 2. The adaptive security experiment of WIBE

It is obvious that CIBE is a special type of WIBE when encrypt under an identity id we encrypt under a pattern $P = \mathsf{id}||*||...||*$ where $|P| = \ell$. In addition, CIBE is also a special type of IBBE, when encrypt under an identity id we encrypt under all descendants of id (including id) using IBBE.

2.4 Hierarchical Identity-Based Encryption

An HIBE scheme consists of four algorithms Setup, KeyDer, Enc, and Dec, which are defined as follows:

- $\mathsf{Setup}(1^\lambda, \ell)$:The setup algorithm takes as input a security parameter 1^λ and maximum hierarchical depth ℓ. It outputs a master key MK and public parameter PP.
- $\mathsf{KeyDer}(\mathsf{PP}, \mathsf{sk_{id}}|_{\mathsf{k}-1}, \mathsf{id}|_k)$:This algorithm takes as input a secret key $\mathsf{sk}_{\mathsf{id}|_{k-1}}$ of hierarchical identity $\mathsf{id}|_{k-1} = (I_1,...,I_{k-1}) \in \mathcal{I}^{k-1}$, a hierarchical identity $\mathsf{id}|_k = (I_1,...,I_k) \in \mathcal{I}^k$ and the public parameter PP. Note that $\mathsf{sk}_{id|_0} = \mathsf{MK}$. It outputs a secret key $\mathsf{sk}_{\mathsf{id}|_k}$ for $\mathsf{id}|_k$.
- $\mathsf{Enc}(\mathsf{id}|_k, \mu, \mathsf{PP})$. The encryption algorithm takes as input a hierarchical identity $\mathsf{id}|_k = (I_1, ..., I_k) \in I^k$, a message μ, and public parameters PP. It outputs a ciphertext $c_{\mathsf{id}|k}$.
- $\mathsf{Dec}(c_{\mathsf{id}|_k}, \mathsf{sk}_{\mathsf{id}|_k}, \mathsf{PP})$:The decryption algorithm takes as input a ciphertext $c_{\mathsf{id}|_k}$, a private key $\mathsf{sk}_{\mathsf{id}|_k}$, and public parameters PP. It outputs a message μ or $\perp$.

Correctness. The correctness of HIBE is defined as follows:
For all $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\mathsf{id}|_{k_0}$, $\mathsf{id}'|_{k_1}$, $\mathsf{Dec}(\mathsf{Enc}(\mathsf{id}'|_{k_1}, \mu, \mathsf{PP}), \mathsf{sk}_{\mathsf{id}'|_{k_1}}, \mathsf{PP}) = \mu$, where $\mathsf{id}|_{k_0}$ is a prefix of $\mathsf{id}'|_{k_1}$ $\mathsf{sk}_{\mathsf{id}|_{k_0}} \leftarrow \mathsf{KeyDer}(\mathsf{PP}, \mathsf{MK}, \mathsf{id}|_{k_0})$ and $\mathsf{sk}_{\mathsf{id}'|_{k_1}} \leftarrow \mathsf{KeyDer}(\mathsf{PP}, \mathsf{sk}_{\mathsf{id}|_{k_0}}, \mathsf{id}'|_{k_1})$.
Adaptive Security: For any PPT adversary $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\cdot)$ such that the following holds:

$$Adv_{\mathcal{A}}^{\mathsf{IND\text{-}HID\text{-}CPA}} = |\Pr[\mathsf{IND\text{-}HID\text{-}CPA}(\mathcal{A}) = 1] - \tfrac{1}{2}| \leq \mathsf{negl}(\lambda)$$

where $\mathsf{IND\text{-}HID\text{-}CPA}(\mathcal{A})$ is shown in Figure 3.

Experiment IND-HID-CPA($\mathcal{A}$) :

1. $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mu_0, \mu_1, \mathsf{id}^*) \leftarrow \mathcal{A}^{\mathsf{KeyDer}(\mathsf{MK},\cdot)}(\mathsf{PP})$ where $|\mu_0| = |\mu_1|$ and for each query id to $\mathsf{KeyDer}(\mathsf{MK},\cdot)$ we have that id is not a prefix of id^*.
3. $\beta \leftarrow \{0,1\}$
4. $\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}^*, \mu_\beta)$
5. $\beta' \leftarrow \mathcal{A}^{\mathsf{KeyDer}(\mathsf{MK},\cdot)}(\mathsf{PP}, \mathsf{c}^*)$ and for each query id to $\mathsf{KeyDer}(\mathsf{MK},\cdot)$ we have that id is not a prefix of id^*.
6. Output 1 if $\beta = \beta'$ and 0 otherwise.

Fig. 3. The adaptive security experiment of HIBE

2.5 Subset Difference Method

The subset difference (SD) method is a special instance of the subset cover framework introduced by Naor, Naor, and Lotspiech [36] which becomes a general methodology for scalable revocation. There is a complete binary tree $\mathcal{BT}$ with 2^ℓ leaves. In our generic RIBE scheme, we view user identity as a leaf in $\mathcal{BT}$. We define $\mathcal{T}_i$ as a complete binary subtree where its root is node v_i. For two nodes in the tree (v_i,v_j) such that v_i is an ancestor of v_j, a valid subtree $\mathcal{T}_{i,j}$ is defined as $\mathcal{T}_i - \mathcal{T}_j$. A valid subset $S_{i,j}$ is represented by (v_i,v_j) which is defined as the set of leaf nodes that belong to $\mathcal{T}_{i,j}$, i.e. a leaf $u \in S_{i,j}$ iff v_i is an ancestor of u but v_j is not. For a full binary tree $\mathcal{BT}$ and a subset R of leaf nodes, $ST(\mathcal{BT}, R)$ is defined as the Steiner Tree induced by the set R and the root node, that is, the minimal subtree of $\mathcal{BT}$ that connects all the leaf nodes in R and the root node. Specifically, the subset difference method is defined as follows:

- $\mathsf{SD.Setup}(N_{max})$: This algorithm takes as input the maximum number N_{max} of users. Let $N_{max} = 2^\ell$ for simplicity. Let $\mathcal{BT}$ denote a complete binary tree of depth ℓ. The corresponding leaf node in the tree of an identity $\mathsf{id} \in \{0,1\}^\ell$ is the terminal node walking from the root directed by id. For an identity $\mathsf{id} = \mathsf{id}_0||\mathsf{id}_1||...||\mathsf{id}_{\ell-1}$, if id_i is 0, go left, otherwise go right at depth i. Note that the root is at depth 0. We set the identifier of the root as 0, so the identifier of corresponding node of id is $0||\mathsf{id}$. The collection S of SD is the set of all subsets $\{S_{i,j}\}$ where $v_i, v_j \in \mathcal{BT}$ and v_i is an ancestor of v_j.
- $\mathsf{SD.Assign}(\mathcal{BT}, \mathsf{id})$: This algorithm takes as input the tree $\mathcal{BT}$ and an identity $\mathsf{id} \in \{0,1\}^\ell$. Let v_{id} be the corresponding leaf node in $\mathcal{BT}$ of id. Let $(v_{k_0}, v_{k_1}, ..., v_{k_\ell})$ be the path from the root node v_{k_0} to the leaf node $v_{k_\ell} = v_{\mathsf{id}}$ and $(v_{k'_1}, ..., v_{k'_\ell})$ be the nodes just "hanging off" the path, i.e. they are adjacent to the path but not ancestors of v_{id}. It first sets a private set PV_{id} as an empty set. For all $i \in \{k_0, k_1, ..., k_\ell\}$ and $j \in \{k'_1, ..., k'_\ell\}$ where v_i is an ancestor of v_j, it adds the subset $S_{i,j}$ presented by two nodes (v_i,v_j) into PV_{id}. It outputs the private set PV_{id}.
- $\mathsf{SD.Cover}(\mathcal{BT}, R)$: This algorithm takes as input the tree $\mathcal{BT}$ and a revoked set R of users. It first sets a subtree $\mathcal{T}$ as $ST(\mathcal{BT}, R)$, and then it builds a covering set CV_R iteratively by removing nodes from $\mathcal{T}$ until $\mathcal{T}$ consists of just a single node as follows:
  (a) It finds two leaf nodes v_i and v_j in $\mathcal{T}$ where the least-common-ancestor v of v_i and v_j does not contain any other leaf nodes of $\mathcal{T}$ in its subtree. Let v_l and v_k be the two child nodes of v where v_l is an ancestor of v_i and v_k is an ancestor of v_j. If there is only one leaf node left, it makes $v_i = v_j$ to the leaf node, v to be the root of $\mathcal{T}$ and $v_l = v_k = v$.
  (b) If $v_i \neq v_l$, then it adds the subset $S_{l,i}$ to CV_R; Similarly, if $v_j \neq v_k$, it adds the subset $S_{k,j}$ to CV_R.
  (c) It removes from $\mathcal{T}$ all the descendants of v and makes v a leaf node.
  It outputs the covering set $CV_R = \{S_{i,j}\}$.
- $\mathsf{SD.Match}(CV_R, PV_{id})$: This algorithm takes input as a covering set $CV_R = \{S_{i,j}\}$ and a private set $PV_{\mathsf{id}} = S'_{i,j}$. It finds two subsets $S_{i,j}$ and $S'_{i',j'}$ such that $S_{i,j} \in CV_R$, $S'_{i',j'} \in PV_{\mathsf{id}}$, and (v_i=$v_{i'}$) $\wedge$($v_j = v_{j'} \vee v_j$ is a descendant of $v_{j'}$) . If it found two subsets, then it outputs $(S_{i,j}, S'_{i',j'})$. Otherwise, it outputs $\perp$.

We give an example of SD.Assign algorithm and SD.Cover algorithm in Figure 4 and Figure 5 respectively.

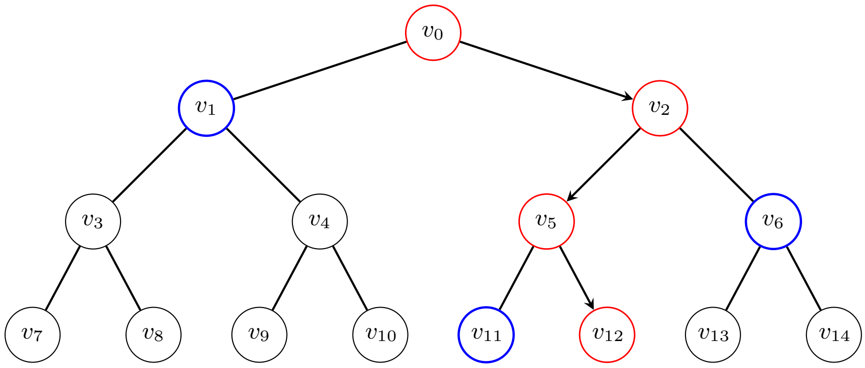

Fig. 4. A private assign to node v_{10} in SD method

For node v_{12}, $PV_{v_{12}}$ is $\{(v_0,v_1),(v_0,v_6),(v_0,v_{11}),(v_2,v_6),(v_2,v_{11}),(v_5,v_{11})\}$

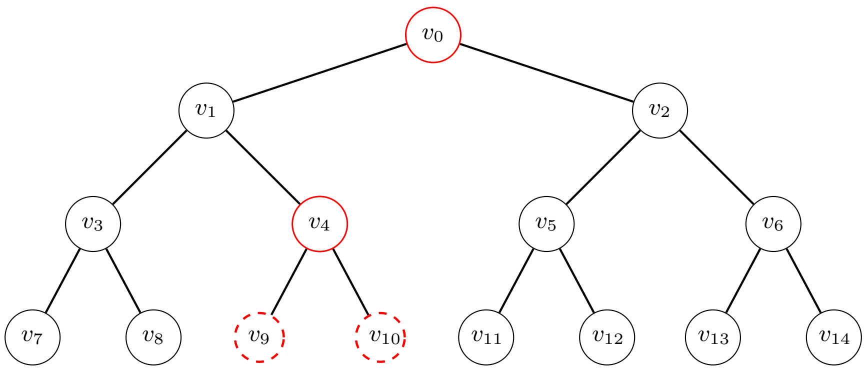

Fig. 5. A cover set for $R = (v_9, v_{10})$ in SD method

$R = \{v_9, v_{10}\}$, $CV = \{(v_0, v_4)\}$.

In order to present our generic RIBE scheme, we first define two functions which encode $CV_{R,\mathsf{T}}$ and PV_{id} to identities respectively. For nodes $v_i = v_{i,0}||...||v_{i,l}$ and $v_j = v_{j,0}||...||v_{j,m}$ where v_i is an ancestor of v_j, we define $H_K : (v_i, v_j) \to \{0,1,2\}^{|v_j|}$ which maps $S_{i,j}$ to the identifier of v_j in $\mathcal{T}_{v_i}$ where $\mathcal{T}_{v_i}$ denotes the complete subtree rooted at v_i. $H_K(v_i, v_j) = v_{j,0}||, ..., ||v_{j,l-1}||2||v_{j,l+1}, ..., ||v_{j,m}$. If $l = 1$, it is $2||v_{j,l+1}, ..., ||v_{j,m}$ and if $m = l + 1$, it is $v_{j,0}||, ..., ||v_{j,l-1}||2||v_{j,m}$. Let $H_E : \{0,1\}^{\ell} \to \{\{0,1,2\}^{\ell+1}\}$ be a function mapping an identity $\mathsf{id} \in \{0,1\}^{\ell}$ to a set of encodings of $S_{i,j} \in PV_{\mathsf{id}}$. Specifically, $H_E(x)$ is defined as follows. Obtain PV_{id} by computing SD.Assign($\mathcal{BT}$, id). Output $\{H_K(S_{i,j})\}_{S_{i,j} \in PV_{\mathsf{id}}}$. From the definition of SD.Assign, we know that there exists an identifier id′ in $H_E(\mathsf{id})$ which is a prefix of $H_K(S_{i,j})$ iff $v_{\mathsf{id}} \in S_{i,j}$. In Figure 5, $CV = (v_0, V_4)$, $H_K(v_0, v_4) = 201$. In figure 4, $PV_{v_{12}}$ is $\{(v_0,v_1),(v_0,v_6),(v_0,v_{11}),(v_2,v_6),(v_2,v_{11}),(v_5,v_{11})\}$ and $H_E(v_{12}) = \{20, 211, 2100, 021, 0200, 0120\}$. There exists "20" which is a prefix of "201" since v_{12} is not revoked. Moreover, $PV_{v_{12}}$ is $\{(v_0,v_2),(v_0,v_3),(v_0,v_{10}),(v_1,v_3),(v_1,v_{10}),(v_4,v_{10})\}$ and $H_E(v_{12}) = \{21, 200, 2011, 020, 0211, 0021\}$. There exists no element in $H_E(v_9)$ which is a prefix of "201" since v_9 is revoked.

3 A Generic Construction of Revocable Identity-Based Encryption

3.1 Definition and Security Model

Similar to the definition in [44], a revocable IBE scheme has seven probabilistic polynomial-time (PPT) algorithms (Setup, KeyGen, KeyUpd, DkGen, Enc, Dec, Revoke) with associated message space $\mathcal{M}$, identity space $\mathcal{ID}$, and time space $\widehat{\mathcal{T}}$.

- $\mathsf{Setup}(1^\lambda, \mathsf{N})$: This algorithm takes as input a security parameter λ 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.
- $\mathsf{KeyGen}(\mathsf{MK}, \mathsf{id}, ST)$: This algorithm takes as input the master secret key MK, an identity id, and the state ST. It outputs a secret key $\mathsf{sk_{id}}$ and an update state ST.
- $\mathsf{KeyUp}(\mathsf{MK}, \mathtt{T}, RL, ST)$: This algorithm takes as input the master secret key MK, a time period $\mathtt{T} \in \widehat{\mathcal{T}}$, the revocation list RL, and the state ST. It outputs a key update $\mathsf{KU_T}$.
- $\mathsf{DkGen}(\mathsf{sk_{id}}, \mathsf{KU_T})$: This algorithm takes as input a secret key $\mathsf{sk_{id}}$ and the key update $\mathsf{KU_T}$. It outputs a decryption $\mathsf{dk_{id,T}}$ or a special symbol $\perp$ indicating that id was revoked.
- $\mathsf{Enc}(\mathsf{PP}, \mathsf{id}, \mathtt{T}, \mu)$: This algorithm takes as input the public parameter PP, an identity id, a time period $\mathtt{T}$ and a message $\mu \in \mathcal{M}$. It outputs a ciphertext c.
- $\mathsf{Dec}(\mathsf{dk_{id,T}}, \mathsf{c})$: This algorithm takes as input a decryption secret key $\mathsf{dk_{id,T}}$ and a ciphertext. It outputs a message $\mu \in \mathcal{M}$.
- $\mathsf{Revoke}(\mathsf{id}, \mathtt{T}, RL)$: This algorithm takes as input an identity id, a revocation time $\mathtt{T} \in \widehat{\mathcal{T}}$ and the revocation list RL. It outputs a revocation list RL.

It satisfies the following conditions:

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

$$\Pr[\mathsf{Dec}(\mathsf{dk_{id,T}}, \mathsf{Enc}(\mathsf{PP}, \mathsf{id}, \mathtt{T}, \mu)) = \mu] \geq 1 - \mathsf{negl}(\lambda)$$

  where $\mathsf{sk_{id}} \leftarrow \mathsf{KeyGen}(\mathsf{MK}, \mathsf{id}, ST)$, $\mathsf{KU_T} \leftarrow \mathsf{KeyUp}(\mathsf{MK}, \mathtt{T}, RL, ST)$ and $\mathsf{dk_{id,T}} \leftarrow \mathsf{DkGen}(\mathsf{sk_{id}}, \mathsf{KU_T})$.
- **Adaptive Security:** For any PPT adversary $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\cdot)$ such that the advantage of $\mathcal{A}$ satisfies:

$$Adv_{\mathcal{A}}^{\mathsf{IND\text{-}RID\text{-}CPA}} = |\Pr[\mathsf{IND\text{-}RID\text{-}CPA}(\mathcal{A}) = 1] - \tfrac{1}{2}| \leq \mathsf{negl}(\lambda)$$

  where $\mathsf{IND\text{-}RID\text{-}CPA}(\mathcal{A})$ is shown is Figure 6. Note that the experiment defined in Figure 6 captures decryption key exposure attack.

3.2 Construction

Let ($\mathsf{CIBE.Setup}, \mathsf{CIBE.Enc}, \mathsf{CIBE.KeyGen}, \mathsf{CIBE.Dec}$) be an CIBE scheme with $\mathcal{ID} = \{0,1,2\}^{[\ell+1, 2\ell+1]}$ and ($\mathsf{HIBE.Setup}, \mathsf{HIBE.Enc}, \mathsf{HIBE.KeyDer}, \mathsf{HIBE.Dec}$) be a two-level HIBE scheme where the element identity is in $\{0,1\}^\ell$. We assume the HIBE scheme and the CIBE scheme have the same plaintext space $\mathcal{M}$ which is finite and forms a group with the group operation "$+$".

Utilizing the above primitives, we will show how to construct a generic RIBE scheme Π = (Setup,KeyGen,KeyUp,DkGen,$\mathsf{Encrypt}$,$\mathsf{Decrypt}$,Revoke) as follows. In our RIBE scheme, the plaintext space is $\mathcal{M}$ and identity space is $\{0,1\}^\ell$. Moreover, we assume the time period space $\widehat{\mathcal{T}}$ is a subset of the identity space, i.e. $\widehat{\mathcal{T}} \subseteq \{0,1\}^\ell$. More specifically, our RIBE scheme is shown as follows:

- $\mathsf{Setup}(1^\lambda, N_{max})$: Run $\mathsf{HIBE.Setup}(1^\lambda, 2) \to (\mathsf{HPP}, \mathsf{HMK})$ and $\mathsf{CIBE.Setup}(1^\lambda) \to (\mathsf{CPP}, \mathsf{CMK})$. $\mathsf{SD.Setup}(1^\lambda, N_{max})$. Output $\mathsf{MK} = \mathsf{HMK}$, an empty revocation list RL, a secret state ST = CMK and public parameter $\mathsf{PP} = (\mathsf{HPP}, \mathsf{CPP})$.
- $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MK}, \mathsf{id})$: Parse PP as $(\mathsf{HPP}, \mathsf{CPP})$, and output $\mathsf{hsk_{id}} \leftarrow \mathsf{HIBE.KeyDer}(\mathsf{HPP}, \mathsf{HMK}, \mathsf{id})$.

Experiment IND-RID-CPA($\mathcal{A}$) :

1. $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{N})$
2. $(\mu_0, \mu_1, \mathsf{id}^*, \mathtt{T}^*) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MK},\cdot),\mathsf{KeyUp}(\mathsf{MK},\cdot,RL,ST),\mathsf{DkGen}(\cdot,\cdot),\mathsf{Revoke}(\cdot,\cdot)}(\mathsf{PP})$ where $|\mu_0| = |\mu_1|$
3. $\beta \leftarrow \{0,1\}$
4. $\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{id}^*, \mathtt{T}^*, \mu_\beta)$
5. $\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MK},.),\mathsf{KeyUp}(\mathsf{MK},\cdot,RL,ST),\mathsf{DkGen}(\cdot,\cdot),\mathsf{Revoke}(\cdot,\cdot)}(\mathsf{PP}, \mathsf{c}^*)$.
6. Output 1 if $\beta = \beta'$ and 0 otherwise.

The following restriction must hold:

- $\mathsf{KeyUp}(\mathsf{MK},\cdot,RL,ST)$ and $\mathsf{Revoke}(\cdot,\cdot)$ 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 $\mathsf{Revoke}(\cdot,\cdot)$ cannot be queried at time $\mathtt{T}$ if $\mathsf{KeyUp}(\mathsf{MK},\cdot,RL,ST)$ was queried on time $\mathtt{T}$.
- If $\mathsf{KeyGen}(\mathsf{MK},\cdot)$ was queried on identity id^*, then $\mathsf{Revoke}(\mathsf{id}^*,\mathtt{T})$ must be queried for some $\mathtt{T} \leq \mathtt{T}^*$, i.e. $(\mathsf{id}^*, \mathtt{T})$ must be on revocation list RL when $\mathsf{KeyUp}(\mathsf{MK},\cdot,RL,ST)$ is queried on $\mathtt{T}^*$.
- $\mathsf{DkGen}(\mathsf{id}^*, \mathtt{T}^*)$ cannot be queried.

Fig. 6. The adaptive security experiment of revocable IBE

- $\mathsf{KeyUp}(\mathsf{PP}, ST, RL, \mathtt{T})$: If there exists $(\mathsf{id}',\mathtt{T}') \in RL$ for some $\mathtt{T}' \leq \mathtt{T}$, add the identifier of id' in $\mathcal{BT}$ to R. Then, obtain $CV_{R,\mathtt{T}} = \{S_{i,j}\}$ by running $\mathsf{SD.Cover}(\mathcal{BT}, R)$. For each $S_{i,j} \in CV_{R,\mathtt{T}}$, compute $\mathsf{csk}_{S_{i,j}} \leftarrow \mathsf{CIBE.KeyGen}(\mathsf{CPP}, \mathsf{CMK}, \mathtt{T}, H_K(S_{i,j}))$. Output updated key $\mathsf{KU}_\mathtt{T} = \{S_{i,j}, \mathsf{csk}_{S_{i,j}}\}_{S_{i,j} \in CV_{R,\mathtt{T}}}$.
- $\mathsf{Enc}(\mathsf{PP}, \mathsf{id}, \mathtt{T}, \mu)$: Parse PP as HPP and CPP. Randomly choose μ_0, μ_1 with the condition that $\mu = \mu_0 + \mu_1$. Compute $c_0 \leftarrow \mathsf{HIBE.Enc}(\mathsf{HPP}, \mathsf{id}||\mathtt{T}, \mu_0)$. For each $\mathsf{id}' \in H_E(\mathsf{id})$, compute $c_{\mathsf{id}'} \leftarrow \mathsf{CIBE.Enc}(\mathsf{CPP}, \mathtt{T}||\mathsf{id}', \mu_1)$. Output $c = \{c_0, \mathtt{T}, \{\mathsf{id}', c_{\mathsf{id}'}\}_{\mathsf{id}' \in H_E(\mathsf{id})}\}$.
- $\mathsf{DkGen}(\mathsf{sk}_\mathsf{id}, \mathsf{KU}_\mathtt{T})$: Parse $\mathsf{KU}_\mathtt{T}$ as $\{S_{i,j}, \mathsf{csk}_{S_{i,j}}\}_{S_{i,j} \in CV_{R,\mathtt{T}}}$. Obtain PV_id by computing $\mathsf{SD.Assign}$ $(\mathcal{BT}, \mathsf{id})$. If $\mathsf{SD.Match}(CV_{R,\mathtt{T}}, PV_\mathsf{id})$ outputs $(S_{i,j}, S'_{i',j'})$, fetch $\mathsf{csk}_{S_{i,j}}$; otherwise, output $\perp$ and abort. Compute $\mathsf{hsk}_{\mathsf{id},\mathtt{T}} \leftarrow \mathsf{HIBE.KeyDer}(\mathsf{HPP}, \mathsf{hsk}_\mathsf{id}, \mathtt{T})$. Output $\mathsf{dk}_{\mathsf{id},\mathtt{T}} = (\mathsf{hsk}_{\mathsf{id},\mathtt{T}}, \mathtt{T}, S_{i,j}, \mathsf{csk}_{S_{i,j}})$.
- $\mathsf{Dec}(\mathsf{PP}, c, \mathsf{sk}_{\mathsf{id},\mathtt{T}})$: Parse $\mathsf{sk}_{\mathsf{id},\mathtt{T}}$ as $(\mathsf{hsk}_{\mathsf{id},\mathtt{T}}, \mathtt{T}, S_{i,j}, \mathsf{csk}_{S_{i,j}})$. Parse c as $c_0, \mathtt{T}', \{\mathsf{id}', c_{\mathsf{id}'}\}_{\mathsf{id}' \in H_E(\mathsf{id})}$. if $\mathtt{T} \neq \mathtt{T}'$, abort; Otherwise, find the identifier id' which is a prefix of $H_K(S_{i,j})$, compute $\mu_1 \leftarrow \mathsf{CIBE.Dec}(\mathsf{CPP}, \mathsf{csk}_{S_{i,j}}, c_{\mathsf{id}'})$ and $\mu_0 \leftarrow \mathsf{HIBE.Dec}(\mathsf{HPP}, \mathsf{hsk}_{\mathsf{id},\mathtt{T}}, c_0)$. Output $\mu = \mu_0 + \mu_1$.
- $\mathsf{Revoke}(ST, RL, \mathtt{T}, \mathsf{id})$: It adds $(\mathsf{id},\mathtt{T})$ to RL and outputs the updated revocation list RL.

3.3 Security Analysis

Theorem 1. *The revocable IBE is adaptive-ID (selective-ID) secure with decryption key exposure resilience if the underlying CIBE scheme and the underlying two-level HIBE scheme are adaptive-ID (selective-ID) secure.*

Proof. We will prove the adaptive-ID security and the proof for selective-ID security is exactly the same. For any PPT adversary against the adaptive-ID security with DKER of revocable IBE, we can construct a PPT algorithm $\mathcal{B}$ against the adaptive-ID security of the underlying CIBE or HIBE scheme. $\mathcal{B}$ randomly guesses an adversarial type among the following two types which are mutually exclusive and cover all possibilities:

1. Type-1 adversary: $\mathcal{A}$ issues a secret key query for id^* hence id^* has to be revoked before $\mathtt{T}^*$.
2. Type-2 adversary: $\mathcal{A}$ does not issue a secret key query for id^*.

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

Type-1 adversary: We will show that if adversary $\mathcal{A}_1$ makes a type-1 attack successfully, there exists an adversary $\mathcal{B}_1$ breaking the multi-identity adaptive security of CIBE defined in Figure 1. $\mathcal{B}_1$ proceeds as follows:

- Setup: $\mathcal{B}_1$ obtains a public parameter CPP from its challenger. It generates $(\mathsf{HPP}, \mathsf{HMK}) \leftarrow \mathsf{HIBE.Setup}(1^\lambda, 2)$ and sends $(\mathsf{HPP},\mathsf{CPP})$ to $\mathcal{A}_1$. $\mathcal{B}_1$ keeps HMK as the master secret key and initial revocation list RL and an identifier set R as empty set.
- KeyGen: When receiving a secret key query for id, if there exists a record of $(\mathsf{id},\mathsf{hsk}_{\mathsf{id}})$ return $\mathsf{hsk}_{\mathsf{id}}$. Otherwise, $\mathcal{B}_1$ generates the secret key normally by running $\mathsf{hsk}_{\mathsf{id}} \leftarrow \mathsf{HIBE.KeyDer}(\mathsf{HMK}, \mathsf{id})$ and record $(\mathsf{id}, \mathsf{hsk}_{\mathsf{id}})$.
- Revoke: $\mathcal{B}_1$ receives $(\mathsf{id},\mathtt{T})$ from $\mathcal{A}_1$, and adds $(\mathsf{id}, \mathtt{T})$ to RL.
- KeyUp: Upon receiving $\mathtt{T}$, for all $(\mathsf{id}',\mathtt{T}') \in RL$ where $\mathtt{T}' \leq \mathtt{T}$, add the identifier of id' in $\mathcal{BT}$ to R. Then, obtain $CV_{R,\mathtt{T}} = \{S_{i,j}\}$ by running $\mathsf{SD.Cover}(\mathcal{BT}, R)$. For each $S_{i,j} \in CV_{R,\mathtt{T}}$, compute $H_K(S_{i,j})$. Query secret keys for $\{\mathtt{T}||H_K(S_{i,j})\}_{S_{i,j}\in CV_{R,\mathtt{T}}}$. Output $\mathsf{KU}_{\mathtt{T}} = \{S_{i,j}, \mathsf{csk}_{S_{i,j}}\}_{S_{i,j}\in CV_{R,\mathtt{T}}}$.
- DkGen:When receiving $(\mathsf{id}, \mathtt{T})$, if there exists a record $(\mathsf{id},\mathsf{hsk}_{\mathsf{id}})$ fetch $\mathsf{hsk}_{\mathsf{id}}$. Otherwise, $\mathcal{B}_1$ can normally run the $\mathsf{HIBE.KeyDer}$ algorithm with HMK and record $(\mathsf{id},\mathsf{hsk}_{\mathsf{id}})$. Note that $\mathsf{KeyUp}(\mathtt{T})$ has been queried before. $\mathcal{B}_1$ outputs $\mathsf{DkGen}(\mathsf{hsk}_{\mathsf{id}},\mathsf{KU}_{\mathtt{T}})$.
- $\mathsf{Challenge}$: $\mathcal{A}_1$ outputs an identity id^*, a time period $\mathtt{T}^*$ and two plaintexts μ_0, μ_1 with the same length. $\mathcal{B}_1$ randomly samples $\mu \leftarrow \mathcal{M}$ and sends $\{\mathtt{T}^*||\mathsf{id}'\}_{\mathsf{id}'\in H_E(\mathsf{id}^*)}$ as challenger identities and $\mu_0' = \mu_0 - \mu$ and $\mu_1' = \mu_1 - \mu$ as the challenge plaintexts. The challenger randomly chooses a challenge bit β and sends the challenge ciphertexts $\{c_{\mathsf{id}'} \leftarrow \mathsf{CIBE.Enc}(\mathsf{CPP}, \mathtt{T}||\mathsf{id}', \mu_\beta)\}_{\mathsf{id}'\in H_E(\mathsf{id}^*)}$ to $\mathcal{B}_1$. $\mathcal{B}_1$ then computes $c_0^* = \mathsf{HIBE.Enc}(\mathsf{HPP}, \mathsf{id}^*||\mathtt{T}^*, \mu)$ and sends $c^* = (c_0^*, \mathtt{T}^*, \{\mathsf{id}', c_{\mathsf{id}'}\}_{\mathsf{id}'\in H_E(\mathsf{id}^*)})$ to $\mathcal{A}_1$.
- Guess: $\mathcal{A}_1$ outputs a guess bit β' and $\mathcal{B}_1$ set β' as its guess.

Due to id^* has been revoked at or before $\mathtt{T}^*$, v_{id^*} is not covered by $CV_{R,\mathtt{T}}$, i.e. $\mathsf{SD.Match}(CV_{R,\mathtt{T}}, PV_{\mathsf{id}^*})$ outputs $\perp$. The property of the encoding function H_E and H_K guarantees that no one identifier id' in $H_E(\mathsf{id}^*)$ is a prefix of $H_K(S_{i,j})$ where $S_{i,j} \in CV_{R,\mathtt{T}}$. So $\mathcal{B}_1$ does not ask any secret key queries for id where some challenge identity is a prefix of id. $\mathcal{B}_1$ perfectly simulates $\mathcal{A}_1$'s view so that $\mathcal{B}_1$'s challenge bit is also $\mathcal{A}_1$'s challenge bit. $\mathcal{B}_1$ just forwards $\mathcal{A}_1$'s guess so the probability that $\mathcal{B}_1$ wins in multi-identity adaptive security game of CIBE scheme is equal to the probability that $\mathcal{A}_1$ wins in adaptive-ID security with decryption key exposure game of RIBE scheme.

Type-2 adversary: If there exists an adversary $\mathcal{A}_2$ who makes a type-2 attack successfully, we can construct an adversary $\mathcal{B}_2$ breaking adaptive-ID security of the underlying HIBE scheme. $\mathcal{B}_2$ proceeds as follows:

- Setup: $\mathcal{B}_2$ obtains a public parameter HPP from its challenger. It generates $(\mathsf{CPP}, \mathsf{CMK}) \leftarrow \mathsf{CIBE.Setup}(1^\lambda)$ and sends $(\mathsf{HPP},\mathsf{CPP})$ to $\mathcal{A}_2$. $\mathcal{B}_2$ keeps CMK as the state.
- KeyGen: When receiving a secret key query for id, $\mathcal{B}_2$ just forwards the secret key query to its challenger and sends the challenger's response to $\mathcal{A}_2$.
- Revoke: $\mathcal{B}_2$ receives $(\mathsf{id},\mathtt{T})$ from $\mathcal{A}_2$, and adds $(\mathsf{id}, \mathtt{T})$ to RL.
- KeyUp: When $\mathcal{A}_2$ makes a key update query for time $\mathtt{T}$, $\mathcal{B}_2$ generates the updated key normally by using CMK.
- DkGen: Upon receiving $(\mathsf{id},\mathtt{T})$. $\mathcal{B}$ queries secret key oracle for $\mathsf{id}||\mathtt{T}$ and obtains $\mathsf{hsk}_{\mathsf{id}||\mathtt{T}}$. Note that $\mathsf{KeyUp}(\mathtt{T})$ has been queried. Then runs the DkGen algorithm normally.
- $\mathsf{Challenge}$: $\mathcal{A}_2$ outputs a challenge identity id^*, a time period $\mathtt{T}^*$ and two plaintexts μ_0 and μ_1 with the same length. $\mathcal{B}_1$ randomly samples $\mu \leftarrow \mathcal{M}$ and sends $\mathsf{id}^*||\mathtt{T}^*$, $\mu_0' = \mu_0 - \mu$ and $\mu_1' = \mu_1 - \mu$ to its challenger. $\mathcal{B}_1$ receives the challenge ciphertext $c_0^* = \mathsf{HIBE.Enc}(\mathsf{HPP}, \mathsf{id}^*||\mathtt{T}^*, \mu_\beta')$ where β is $\mathcal{B}_2$'s challenge bit chosen randomly by its challenger. For each $\mathsf{id}' \in H_E(\mathsf{id}^*)$, compute $c_{\mathsf{id}'} \leftarrow \mathsf{CIBE.Enc}(\mathsf{CPP}, \mathtt{T}||\mathsf{id}', \mu_1)$ and sends $c = \{c_0^*, \mathtt{T}^*, \{\mathsf{id}', c_{\mathsf{id}'}\}_{\mathsf{id}'\in H_E(\mathsf{id}^*)}\}$ to $\mathcal{A}_2$.
- Guess: $\mathcal{A}_2$ outputs a guess bit β' and $\mathcal{B}_2$ sets β' as its guess.
  For the KeyUp oracle, $\mathcal{B}_2$ can respond by itself because it has the state. For the KeyGen oracle, $\mathcal{A}_2$ never requests secret key for the challenge identity id^*; And $\mathsf{DkGen}(\mathsf{id}^*, \mathtt{T}^*)$ is never queried, so $\mathcal{B}_2$ never requests secret keys for $\mathsf{id}^*||\mathtt{T}^*$ or its ancestors. $\mathcal{B}_2$ perfectly simulates $\mathcal{A}_2$'s view so that $\mathcal{B}_2$'s challenge bit is also $\mathcal{A}_2$'s challenge bit. $\mathcal{B}_2$ just forwards $\mathcal{A}_2$'s guess so the probability that $\mathcal{B}_2$ wins in adaptive-ID security game of HIBE scheme is equal to the probability that $\mathcal{A}_2$ wins in adaptive-ID security with decryption key exposure game of RIBE scheme.

When we put the results for two types of adversary together, we can conclude that the revocable IBE is adaptive-ID secure if both the underlying IBE and HIBE schemes are adaptive-ID secure.

3.4 Extensions

Layered Subset Difference. In our generic RIBE construction, the ciphertext and update key are $O(\ell^2)$ CIBE ciphertexts plus a HIBE ciphertext and $2r$ CIBE private keys respectively where ℓ is the bit length of identity and r is the number of revoked users. We can use layered subset difference (LSD) method [24] to reduce the ciphertext size. If we replace the SD algorithms by LSD algorithms in our generic RIBE construction, the ciphertext and the update key are $O(\ell^{1.5})$ CIBE ciphertexts plus a HIBE ciphertext and $4r$ CIBE private keys respectively.

Constant Ciphertext. Due to CIBE is a special type of IBBE, we can directly replace CIBE by IBBE. The ciphertext in this generic construction consists of $O(\ell^2)/O(\ell^{1.5})$ IBBE ciphertexts plus a HIBE ciphertext if we use SD/LSD method. In addition, we can replace all $O(\ell^2)/O(\ell^{1.5})$ IBBE ciphertexts c_i encrypted under set S_i by one IBBE ciphertext c encrypted under a set $S = \cup S_i$ since all c_i encrypt the same plaintext. So we can reduce the ciphertext to be one IBBE ciphertexts and one HIBE ciphertext.

Server-Aided RIBE. In server-aided model, there is a semi-honest server without any secret key information that takes almost all the workload on users. The server is curious but honestly performs the procedure. More specifically, the server partially decrypts the ciphertexts using the key update and leaves less decryption task to users. It is easy to convert our scheme to be server-aided, given the key update $\mathsf{KU}_{\mathtt{T}} = \{S_{i,j}, \mathsf{csk}_{S_{i,j}}\}_{S_{i,j} \in CV_{R,\mathtt{T}}}$ and a ciphertext $c = \{c_0, \mathtt{T}, \{\mathsf{id}', c_{\mathsf{id}'}\}_{\mathsf{id}' \in H_E(\mathsf{id})}\}$, the sever first computes $PV_{\mathsf{id}} \leftarrow \mathsf{SD.Assign}(\mathcal{BT}, \mathsf{id})$. If $\mathsf{SD.Match}(CV_{R,\mathtt{T}}, PV_{\mathsf{id}})$ outputs $(S_{i,j}, S'_{i',j'})$, fetch $\mathsf{csk}_{S_{i,j}}$ from $\mathsf{KU}_{\mathtt{T}}$ and compute $\mu_1 \leftarrow \mathsf{CIBE.Dec}(\mathsf{CPP}, \mathsf{csk}_{S_{i,j}}, c_{\mathsf{id}'})$ where id' is a prefix of $H_E(S_{i,j})$. Finally, the sever sends $(\mathsf{c}_0, \mathtt{T}, \mu_1)$ as the transformed ciphertext to the receiver. The receiver only needs to operate the key derive and decryption algorithm of underlying HIBE scheme. The receiver does not need to communicate with KGC in every key update.

3.5 Instantiation

If we instantiate the underlying two-level HIBE scheme with BBG-HIBE [8] and the underlying IBBE scheme with a IBBE scheme with constant size public parameter, ciphertext and private key presented in [26], we can obtain a selectively secure RIBE with constant public parameter, ciphertext, private key and $O(r)$ key update. To the best of our knowledge, it is the first RIBE scheme that realizes constant size of public parameter, ciphertext, private key and $O(r)$ number of key update simultaneously. Additionally, if we instantiate the underlying two-level HIBE scheme with BBG-HIBE [8] and the underlying WIBE scheme with BBG-WIBE scheme [1], we can obtain an adaptively secure RIBE in the random oracle where the sizes of ciphertexts and key update are $O(\ell^{2.5})$ and $O(r)$ respectively (using LSD method). The ciphertext-policy attribute-based encryption presented in [38] implies WIBE and combining the result in [18] that HIBE can be constructed from IBE, we can obtain a RIBE scheme based on RSA.

4 Conclusion

In this paper, we presented a new primitive called IBE with identity delegation (CIBE) where an identity secret key can decrypt ciphertexts encrypted under its ancestors. CIBE is a special type of WIBE and IBBE and can be constructed from IBE in a black-box way. We then proposed a generic RIBE scheme via subset difference method using CIBE and two-level HIBE as building blocks. In our generic RIBE scheme, the key update consists of $O(r)$ CIBE private keys and ciphertext consists of $O(\ell^2)$ CIBE ciphertexts and one HIBE ciphertext. The ciphertext size can be reduced to $O(\ell^{1.5})$ by using layered subset difference method. Moreover, the generic RIBE scheme can be converted to a server-aided RIBE scheme and be instantiated efficiently. We can reduce the ciphertext size using IBBE and the instantiated RIBE scheme has constant-size public parameter, ciphertext, private key and $O(r)$ key update. We can obtain RIBE based on RSA assumption if we instantiate the underlying buildings based on RSA.

References

1. Michel Abdalla, Dario Catalano, Alexander W. Dent, John Malone-Lee, Gregory Neven, and Nigel P. Smart. Identity-based encryption gone wild. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 300–311, 2006.
2. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.
3. Daniel Apon, Xiong Fan, and Feng-Hao Liu. Fully-secure lattice-based IBE as compact as PKE. *IACR Cryptology ePrint Archive*, 2016:125, 2016.
4. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93,*, pages 62–73. ACM, 1993.
5. Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In *CCS 2008*, pages 417–426. ACM, 2008.
6. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.
7. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, 2004.
8. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT 2005*, pages 440–456, 2005.
9. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, pages 213–229, 2001.
10. Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. *IACR Cryptology ePrint Archive*, 2007:177, 2007.
11. Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography - PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517. Springer, 2010.
12. Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and id-based encryption. In *ASIACRYPT 2016*, volume 10032 of *Lecture Notes in Computer Science*, pages 404–434, 2016.
13. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.
14. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Khoa Nguyen. Revocable identity-based encryption from lattices. In *ACISP 2012*, volume 7372 of *Lecture Notes in Computer Science*, pages 390–403. Springer, 2012.
15. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *Cryptography and Coding*, pages 360–363, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
16. Hui Cui, Robert H. Deng, Yingjiu Li, and Baodong Qin. Server-aided revocable attribute-based encryption. In *Computer Security - ESORICS 2016*, volume 9879 of *Lecture Notes in Computer Science*, pages 570–587. Springer, 2016.
17. Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In *ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2007.
18. Nico Döttling and Sanjam Garg. From selective IBE to full IBE and selective HIBE. In *TCC 2017*, volume 10677 of *Lecture Notes in Computer Science*, pages 372–408. Springer, 2017.
19. Nico Döttling and Sanjam Garg. Identity-based encryption from the diffie-hellman assumption. In *CRYPTO 2017*, volume 10401 of *Lecture Notes in Computer Science*, pages 537–569. Springer, 2017.
20. Keita Emura, Jae Hong Seo, and Taek-Young Youn. Semi-generic transformation of revocable hierarchical identity-based encryption and its DBDH instantiation. *IEICE Transactions*, 99-A(1):83–91, 2016.
21. Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, pages 445–464, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
22. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206. ACM, 2008.
23. Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.
24. Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60, 2002.

25. Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481, 2002.
26. Liang Hu, Zheli Liu, and Xiaochun Cheng. Efficient identity-based broadcast encryption without random oracles. *JCP*, 5(3):331–336, 2010.
27. Yuu Ishida, Junji Shikata, and Yohei Watanabe. Cca-secure revocable identity-based encryption schemes with decryption key exposure resistance. *IJACT*, 3(3):288–311, 2017.
28. Shuichi Katsumata, Takahiro Matsuda, and Atsushi Takayasu. Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance. In *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 441–471, 2019.
29. Kwangsu Lee. Revocable hierarchical identity-based encryption with adaptive security. *IACR Cryptology ePrint Archive*, 2016:749, 2016.
30. Kwangsu Lee. A generic construction for revocable identity-based encryption with subset difference methods. *IACR Cryptology ePrint Archive*, 2019:798, 2019.
31. Kwangsu Lee, Dong Hoon Lee, and Jong Hwan Park. Efficient revocable identity-based encryption via subset difference methods. *Des. Codes Cryptography*, 85(1):39–76, 2017.
32. Kwangsu Lee and Seunghwan Park. Revocable hierarchical identity-based encryption with shorter private keys and update keys. *Des. Codes Cryptography*, 86(10):2407–2440, 2018.
33. Benoît Libert and Damien Vergnaud. Adaptive-id secure revocable identity-based encryption. In *CT-RSA 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2009.
34. Xuecheng Ma and Dongdai Lin. A generic construction of revocable identity-based encryption. *IACR Cryptology ePrint Archive*, 2019:299, 2019.
35. Xianping Mao, Junzuo Lai, Kefei Chen, Jian Weng, and Qixiang Mei. Efficient revocable identity-based encryption from multilinear maps. *Security and Communication Networks*, 8(18):3511–3522, 2015.
36. Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.
37. Khoa Nguyen, Huaxiong Wang, and Juanyang Zhang. Server-aided revocable identity-based encryption from lattices. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings*, volume 10052 of *Lecture Notes in Computer Science*, pages 107–123, 2016.
38. Vanga Odelu, Ashok Kumar Das, Muhammad Khurram Khan, Kim-Kwang Raymond Choo, and Minho Jo. Expressive CP-ABE scheme for mobile devices in iot satisfying constant-size keys and ciphertexts. *IEEE Access*, 5:3273–3283, 2017.
39. Seunghwan Park, Dong Hoon Lee, and Kwangsu Lee. Revocable hierarchical identity-based encryption from multilinear maps. *CoRR*, abs/1610.07948, 2016.
40. Seunghwan Park, Kwangsu Lee, and Dong Hoon Lee. New constructions of revocable identity-based encryption from multilinear maps. *IEEE Trans. Information Forensics and Security*, 10(8):1564–1577, 2015.
41. Baodong Qin, Robert H. Deng, Yingjiu Li, and Shengli Liu. Server-aided revocable identity-based encryption. In *Computer Security - ESORICS 2015*, volume 9326 of *Lecture Notes in Computer Science*, pages 286–304, 2015.
42. Geumsook Ryu, Kwangsu Lee, Seunghwan Park, and Dong Hoon Lee. Unbounded hierarchical identity-based encryption with efficient revocation. In *WISA 2015*, volume 9503 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2015.
43. Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.
44. Jae Hong Seo and Keita Emura. Revocable identity-based encryption revisited: Security model and construction. In *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, volume 7778 of *Lecture Notes in Computer Science*, pages 216–234. Springer, 2013.
45. Jae Hong Seo and Keita Emura. Revocable hierarchical identity-based encryption. *Theor. Comput. Sci.*, 542:44–62, 2014.
46. Jae Hong Seo and Keita Emura. Revocable hierarchical identity-based encryption via history-free approach. *Theor. Comput. Sci.*, 615:45–60, 2016.
47. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

48. Atsushi Takayasu and Yohei Watanabe. Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance. In *ACISP 2017*, volume 10342 of *Lecture Notes in Computer Science*, pages 184–204. Springer, 2017.
49. Yohei Watanabe, Keita Emura, and Jae Hong Seo. New revocable IBE in prime-order groups: Adaptively secure, decryption key exposure resistant, and with short public parameters. In *CT-RSA 2017*, volume 10159 of *Lecture Notes in Computer Science*, pages 432–449. Springer, 2017.
50. Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, pages 114–127, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
51. Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, pages 619–636, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
52. Shota Yamada. Asymptotically compact adaptively secure lattice ibes and verifiable random functions via generalized partitioning techniques. In *CRYPTO 2017*, volume 10403 of *Lecture Notes in Computer Science*, pages 161–193. Springer, 2017.
53. Jiang Zhang, Yu Chen, and Zhenfeng Zhang. Programmable hash functions from lattices: Short signatures and ibes with small key sizes. In *CRYPTO 2016*, volume 9816 of *Lecture Notes in Computer Science*, pages 303–332. Springer, 2016.