Simplified Revocable Hierarchical Identity-Based Encryption from Lattices

Shixiong Wang[1,4], Juanyang Zhang[2], Jingnan He[3,4], Huaxiong Wang[4], and Chao Li[1]

1 College of Computer, National University of Defense Technology, Changsha, China
2 School of Information Engineering, Ningxia University, Yinchuan, China
3 State Key Laboratory of Information Security, Institute of Information Engineering of Chinese Academy of Sciences, Beijing, China
4 School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore
wsx09@foxmail.com,jyzhang@nxu.edu.cn,hejingnan@iie.ac.cn,
hxwang@ntu.edu.sg,lichao_nudt@sina.com

Abstract. As an extension of identity-based encryption (IBE), revocable hierarchical IBE (RHIBE) supports both key revocation and key delegation simultaneously, which are two important functionalities for cryptographic use in practice. Recently in PKC 2019, Katsumata et al. constructed the first lattice-based RHIBE scheme with decryption key exposure resistance (DKER). Such constructions are all based on bilinear or multilinear maps before their work. In this paper, we simplify the construction of RHIBE scheme with DKER provided by Katsumata et al. With our new treatment of the identity spaces and the time period space, there is only one short trapdoor base in the master secret key and in the secret key of each identity. In addition, we claim that some items in the keys can also be removed due to the DKER setting. Our first RHIBE scheme in the standard model is presented as a result of the above simplification. Furthermore, based on the technique for lattice basis delegation in fixed dimension, we construct our second RHIBE scheme in the random oracle model. It has much shorter items in keys and ciphertexts than before, and also achieves the adaptive-identity security under the learning with errors (LWE) assumption.

Keywords: Lattices, Identity-based encryption, Revocation, Delegation

1 Introduction

Background. Identity-based encryption (IBE), envisaged by Shamir [20] in 1984, is an advanced form of public-key encryption (PKE) where any string such as an email address can be used as a public key. Hierarchical IBE (HIBE), an extension of IBE introduced by Horwitz and Lynn [10] in 2002, further supports a key delegation functionality. Moreover, just as many multi-user cryptosystems, an efficient revocation mechanism is usually necessary and imperative in the (H)IBE setting. The public/private key pair of a system user may need to be removed for various reasons, such as that the user is no longer a legitimate system user, or that the private key is lost or stolen. Designing the revocable IBE (RIBE) or revocable HIBE (RHIBE) turned out to be a challenging problem.

In 2001, Boneh and Franklin [6] proposed a naive solution for RIBE, which requires users to periodically renew their private keys. This solution is too impractical to be used in large-scale system, since for the key generation center (denoted by KGC), the workload grows linearly in the number of users N. Later in 2008, Boldyreva et al. [5] utilized the complete subtree (CS) method of Naor et al. [17] to construct the first scalable RIBE, where KGC's workload is only logarithmic in N. RIBE requires three types of keys: a secret key SK, a key update KU, and a decryption key DK. For each time period t, the KGC broadcasts a key update $\mathsf{KU}_{\mathsf{KGC},\mathsf{t}}$ through a public channel, and only non-revoked identity ID at this time period t can derive a decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ by combining its secret key $\mathsf{SK}_{\mathsf{ID}}$ with the key update $\mathsf{KU}_{\mathsf{KGC,t}}$. In the security model of [5], the adversary only has the access to a secret key reveal oracle and a key update reveal oracle. However, leakage of decryption keys may also happen in practice. In 2013, Seo and Emura [19] introduced a new security notion called decryption key exposure resistance (DKER), and thus refined the security model, where the adversary also has the access to a decryption key reveal oracle. The works in [5] and [19] attracted a lot of followup works, and their RIBE schemes were also extended to RHIBE schemes. Note that before Katsumata et al.'s work [11] in 2019, the constructions of R(H)IBE schemes with DKER are all based on bilinear or multilinear maps, and they rely heavily on the so-called key re-randomization property.

This paper focuses on the lattice-based cryptography, which has faster arithmetic operations and conjectured security against quantum attacks. In 2012, Chen et al. [8] employed Agrawal et al.'s IBE [1] and the CS method [17] to construct the first lattice-based RIBE scheme without DKER. Then in 2017, Takayasu and Watanabe [21] presented a new lattice-based RIBE scheme secure against exposure of a-priori bounded number of decryption keys for every identity. Namely, their scheme only achieves bounded DKER. Later in 2019, Katsumata et al. [11] proposed the first lattice-based R(H)IBE scheme with DKER under the learning with errors (LWE) assumption. Specifically, they provided a generic construction of RIBE with DKER from any RIBE without DKER and two-level HIBE. This result directly implies the first lattice-based RIBE scheme with DKER. Furthermore, they constructed the first lattice-based RHIBE scheme with DKER by further exploiting the algebraic structure of lattices. Since lattices are ill-fit with the so-called key re-randomization property, Katsumata et al. [11] introduced new tools such as leveled ciphertexts, leveled secret keys, leveled decryption keys, and level conversion keys. Therefore, their techniques highly depart from previous works which are based on bilinear or multilinear maps.

Our Contributions and Techniques. In this paper, we manage to simplify the construction of lattice-based RHIBE scheme with DKER in [11]. Specifically, we present two new RHIBE schemes $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$, both of which are based on lattices and achieve DKER. Let $\mathbf{\Pi}_0$ denote the RHIBE scheme with DKER in [11]. Then compared with $\mathbf{\Pi}_0$, our first scheme $\mathbf{\Pi}_1$ has fewer items in the public parameters, secret keys, and key updates. Furthermore, in our second scheme $\mathbf{\Pi}_2$, the items in keys and ciphertexts are much shorter than $\mathbf{\Pi}_0, \mathbf{\Pi}_1$. The scheme $\mathbf{\Pi}_0$ in [11] and our first scheme $\mathbf{\Pi}_1$ are in the standard model, and they both satisfy the selective-identity security, assuming the hardness of the LWE problem. While our second scheme $\mathbf{\Pi}_2$, which is in the random oracle model, achieves the adaptive-identity security under the LWE assumption.

In Figure 1, we show the public parameters PP, the master secret key $\mathsf{SK}_{\mathsf{KGC}}$ (the secret key of KGC), the ciphertext CT, the secret key $\mathsf{SK}_{\mathsf{ID}}$, the key update $\mathsf{KU}_{\mathsf{ID,t}}$, and the decryption key $\mathsf{DK}_{\mathsf{ID,t}}$, together with the description of their items, for the schemes $\mathbf{\Pi}_0$, $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$. In this figure, L is the maximum depth of the hierarchy, and we use $\ell := |\mathsf{ID}|$ to denote the depth of the corresponding ID explicitly in $\mathsf{SK}_{\mathsf{ID}}$, $\mathsf{KU}_{\mathsf{ID,t}}$, $\mathsf{DK}_{\mathsf{ID,t}}$, or implicitly in CT, respectively. In addition, for $n_1, n_2 \in \mathbb{N}$, we set $[n_1, n_2] := \{n_1, n_1+1, \cdots, n_2\}$ if $n_1 \leqslant n_2$, or $[n_1, n_2] := \emptyset$ if $n_1 > n_2$, and then let $[n] := [1, n]$ for $n \in \mathbb{N}$. Figure 1 only provides a brief description of the RHIBE schemes $\mathbf{\Pi}_0, \mathbf{\Pi}_1, \mathbf{\Pi}_2$, and the notations in this figure will be clarified later in this paper when necessary. For example, the notation $\mathsf{BT}_{\mathsf{KGC}}$ (or $\mathsf{BT}_{\mathsf{ID}}$), which denotes a binary tree managed by KGC (or ID), is introduced in Section 2.3. The function $\mathbf{E}(\cdot)$ used in $\mathsf{SK}_{\mathsf{ID}}$ for $\mathbf{\Pi}_0, \mathbf{\Pi}_1$ is described in Section 3, and the functions $\mathbf{P}_1(\cdot), \mathbf{P}_2(\cdot)$ used in $\mathsf{SK}_{\mathsf{ID}}$ for $\mathbf{\Pi}_2$ are defined in Section 4. Actually, Figure 1 is mainly for the comparison, from which we can see that our first scheme $\mathbf{\Pi}_1$ needs fewer items than $\mathbf{\Pi}_0$, and the sizes of items are much smaller in our second scheme $\mathbf{\Pi}_2$. Furthermore, with the help of Figure 1, we can briefly introduce our techniques as follows.

In the RHIBE, each identity $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_\ell)$ at level $\ell \in [L]$ belongs to the hierarchical identity space $\mathcal{ID}_{\mathsf{H}} = (\mathcal{ID})^{\leqslant L} := \bigcup_{i \in [L]} (\mathcal{ID})^i$, where $\mathcal{ID}$ is the element identity space. The KGC, i.e., the key generation center, is the unique level-0 identity. For the construction of our scheme $\mathbf{\Pi}_1$, we introduce another space $\widetilde{\mathcal{ID}}$ such that $\mathcal{ID} \cap \widetilde{\mathcal{ID}} = \emptyset$, $|\mathcal{ID}| = |\widetilde{\mathcal{ID}}|$, and there is

The RHIBE Scheme $\mathbf{\Pi}_0$ in [11]	The items in $\mathbf{\Pi}_0$
1. $\mathsf{PP} = \Big(\boxed{(\mathbf{A}_i)_{i\in[L+1]}},\ (\mathbf{C}_j)_{j\in[L+1]},\ \underline{(\mathbf{u}_k)_{k\in[L]}}\Big),$ $\mathsf{SK}_{\mathsf{KGC}} = \Big(\mathsf{BT}_{\mathsf{KGC}},\ \boxed{(\mathbf{T}_{\mathbf{A}_i})_{i\in[L+1]}}\Big)$ 2. $\mathsf{CT} = \Big(c_0,\ (\mathbf{c}_i)_{i\in[\ell]},\ \mathbf{c}_{L+1}\Big)$ 3. $\mathsf{SK}_{\mathsf{ID}} = \Big(\mathsf{BT}_{\mathsf{ID}},\ (\theta, \mathbf{e}_{\mathsf{ID},\theta})_\theta,\ \underline{(\mathbf{f}_{\mathsf{ID},k})_{k\in[\ell+1,L]}},$ $\boxed{(\mathbf{T}_{[\mathbf{A}_i\|\mathbf{E}(\mathsf{ID})]})_{i\in[\ell+1,L+1]}}\Big)$ 4. $\mathsf{KU}_{\mathsf{ID},\mathsf{t}} = \Big((\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_\theta,\ \underline{(\mathbf{f}_{\mathsf{ID}_{[i]},\mathsf{t},k})_{(i,k)\in[\ell]\times[\ell+1,L]}}\Big)$ 5. $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} = \Big((\mathbf{f}_{\mathsf{ID}_{[i]},\mathsf{t},\ell})_{i\in[\ell-1]},\ \mathbf{d}_{\mathsf{ID},\mathsf{t}},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$	1. $\mathbf{A}_i, \mathbf{C}_j \in \mathbb{Z}_q^{n\times m}$, $\mathbf{u}_k \in \mathbb{Z}_q^n$, $\mathbf{T}_{\mathbf{A}_i} \in \mathbb{Z}^{m\times m}$ 2. $c_0 \in \mathbb{Z}_q,\ \mathbf{c}_i \in \mathbb{Z}_q^{(i+2)m},\ \mathbf{c}_{L+1} \in \mathbb{Z}_q^{(\ell+2)m}$ 3. $\mathbf{e}_{\mathsf{ID},\theta}, \mathbf{f}_{\mathsf{ID},k} \in \mathbb{Z}^{(\ell+1)m}$, $\mathbf{T}_{[\mathbf{A}_i\|\mathbf{E}(\mathsf{ID})]} \in \mathbb{Z}^{(\ell+1)m\times(\ell+1)m}$ 4. $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} \in \mathbb{Z}^{(\ell+2)m}$, $\mathbf{f}_{\mathsf{ID}_{[i]},\mathsf{t},k} \in \mathbb{Z}^{(i+2)m}$ 5. $\mathbf{f}_{\mathsf{ID}_{[i]},\mathsf{t},\ell} \in \mathbb{Z}^{(i+2)m}$, $\mathbf{d}_{\mathsf{ID},\mathsf{t}}, \mathbf{g}_{\mathsf{ID},\mathsf{t}} \in \mathbb{Z}^{(\ell+2)m}$
Our First RHIBE Scheme $\mathbf{\Pi}_1$	The items in $\mathbf{\Pi}_1$
1. $\mathsf{PP} = \Big(\boxed{\mathbf{A}},\ (\mathbf{C}_i)_{i\in[L+1]},\ \underline{\mathbf{u}}\Big),$ $\mathsf{SK}_{\mathsf{KGC}} = \Big(\mathsf{BT}_{\mathsf{KGC}},\ \boxed{\mathbf{T}_{\mathbf{A}}}\Big)$ 2. $\mathsf{CT} = \Big(c_0,\ (\mathbf{c}_i)_{i\in[\ell]},\ \mathbf{c}_{L+1}\Big)$ 3. $\mathsf{SK}_{\mathsf{ID}} = \Big(\mathsf{BT}_{\mathsf{ID}},\ (\theta, \mathbf{e}_{\mathsf{ID},\theta})_\theta,\ \boxed{\mathbf{T}_{[\mathbf{A}\|\mathbf{E}(\mathsf{ID})]}}\Big)$ 4. $\mathsf{KU}_{\mathsf{ID},\mathsf{t}} = \Big((\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_\theta,\ \underline{(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}}\Big)$ 5. $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} = \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$	1. $\mathbf{A}, \mathbf{C}_i \in \mathbb{Z}_q^{n\times m}$, $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m\times m}$ 2. $c_0 \in \mathbb{Z}_q,\ \mathbf{c}_i \in \mathbb{Z}_q^{(i+2)m},\ \mathbf{c}_{L+1} \in \mathbb{Z}_q^{(\ell+2)m}$ 3. $\mathbf{e}_{\mathsf{ID},\theta} \in \mathbb{Z}^{(\ell+1)m}$, $\mathbf{T}_{[\mathbf{A}\|\mathbf{E}(\mathsf{ID})]} \in \mathbb{Z}^{(\ell+1)m\times(\ell+1)m}$ 4. $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} \in \mathbb{Z}^{(\ell+2)m}$, $\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}} \in \mathbb{Z}^{(i+2)m}$ 5. $\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}} \in \mathbb{Z}^{(i+2)m}$, $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \in \mathbb{Z}^{(\ell+2)m}$
Our Second RHIBE Scheme $\mathbf{\Pi}_2$	The items in $\mathbf{\Pi}_2$
1. $\mathsf{PP} = \Big(\mathbf{A},\ \mathbf{B},\ \mathbf{u}\Big),$ $\mathsf{SK}_{\mathsf{KGC}} = \Big(\mathsf{BT}_{\mathsf{KGC}},\ \mathbf{T}_{\mathbf{A}},\ \mathbf{T}_{\mathbf{B}}\Big)$ 2. $\mathsf{CT} = \Big(c_0,\ (\mathbf{c}_{i,1}, \mathbf{c}_{i,2})_{i\in[\ell]},\ \mathbf{c}_{L+1}\Big)$ 3. $\mathsf{SK}_{\mathsf{ID}} = \Big(\mathsf{BT}_{\mathsf{ID}},\ (\theta, \mathbf{e}_{\mathsf{ID},\theta})_\theta,\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})},\ \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})}\Big)$ 4. $\mathsf{KU}_{\mathsf{ID},\mathsf{t}} = \Big((\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_\theta,\ (\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}\Big)$ 5. $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} = \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$	1. $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{T}_{\mathbf{A}}, \mathbf{T}_{\mathbf{B}} \in \mathbb{Z}^{m\times m}$ 2. $c_0 \in \mathbb{Z}_q,\ (\mathbf{c}_{i,1}, \mathbf{c}_{i,2}) \in \mathbb{Z}_q^{2m},\ \mathbf{c}_{L+1} \in \mathbb{Z}_q^{m}$ 3. $\mathbf{e}_{\mathsf{ID},\theta} \in \mathbb{Z}^{m}$, $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})}, \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})} \in \mathbb{Z}^{m\times m}$ 4. $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} \in \mathbb{Z}^{m}$, $\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}} \in \mathbb{Z}^{2m}$ 5. $\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}} \in \mathbb{Z}^{2m}$, $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \in \mathbb{Z}^{m}$

Fig. 1. Comparison of the RHIBE schemes $\mathbf{\Pi}_0, \mathbf{\Pi}_1, \mathbf{\Pi}_2$

a one-to-one correspondence between $\mathsf{id} \in \mathcal{ID}$ and $\widetilde{\mathsf{id}} \in \widetilde{\mathcal{ID}}$. Suppose that in the encryption algorithm, a message M is encrypted under an identity $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_\ell) \in \mathcal{ID}_{\mathsf{H}}$ (and under a time period t). Then from Figure 1, we know that both the schemes $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_1$ will output the ciphertext $\mathsf{CT} = \Big(c_0,\ (\mathbf{c}_i)_{i\in[\ell]},\ \mathbf{c}_{L+1}\Big) \in \mathbb{Z}_q \times (\mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{4m} \times \cdots \times \mathbb{Z}_q^{(\ell+2)m}) \times \mathbb{Z}_q^{(\ell+2)m}$. However, for $\mathbf{\Pi}_0$ the item $\mathbf{c}_i$ in CT is generated from $\mathsf{ID}_{[i]} := (\mathsf{id}_1, \cdots, \mathsf{id}_{i-1}, \mathsf{id}_i)$, while the item $\mathbf{c}_i$ for our $\mathbf{\Pi}_1$ is created from $\widetilde{\mathsf{ID}_{[i]}} := (\mathsf{id}_1, \cdots, \mathsf{id}_{i-1}, \widetilde{\mathsf{id}_i})$. As a consequence, our scheme $\mathbf{\Pi}_1$ only needs one short trapdoor base $\mathbf{T}_{\mathbf{A}}$ (or $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID})]}$) in the secret key $\mathsf{SK}_{\mathsf{KGC}}$ (or $\mathsf{SK}_{\mathsf{ID}}$), and accordingly the matrix $\mathbf{A}$ is used in PP instead of $(\mathbf{A}_i)_{i\in[L+1]}$, shown as in Figure 1. In the security proof, the adversary $\mathcal{A}$ may issue a secret key reveal query on $\mathsf{ID}^*_{[i^*]}$ but not on any $\mathsf{ID}^*_{[j]}$ for $j \in [i^* - 1]$, where ID^* denotes the challenge identity and $i^* \leqslant |\mathsf{ID}^*|$. In this case, the LWE problem instance is used to construct $\mathbf{A}, \mathbf{u}$ in PP and $c_0, \mathbf{c}_{i^*}$ in CT for our scheme $\mathbf{\Pi}_1$. Though without the trapdoor $\mathbf{T}_{\mathbf{A}}$, we are still able to construct $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID}^*_{[i^*]})]}$ in $\mathsf{SK}_{\mathsf{ID}^*_{[i^*]}}$ for the adversary $\mathcal{A}$, since the simulated $\mathbf{c}_{i^*}$ is only related to $\widetilde{\mathsf{ID}^*_{[i^*]}}$, not $\mathsf{ID}^*_{[i^*]}$ itself. The construction of $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID}^*_{[i^*]})]}$ will not succeed, if $\mathbf{c}_{i^*}$ is obtained in the way of the scheme $\mathbf{\Pi}_0$. This is also the reason why $\mathbf{\Pi}_0$ employs $L + 1$ short trapdoor bases $(\mathbf{T}_{\mathbf{A}_i})_{i\in[L+1]}$ in

$\mathsf{SK}_{\mathsf{KGC}}$, and $L+1-\ell$ short trapdoor bases $(\mathbf{T}_{[\mathbf{A}_i|\mathbf{E}(\mathsf{ID})]})_{i\in[\ell+1,L+1]}$ in $\mathsf{SK}_{\mathsf{ID}}$, just as Figure 1 shows. Similarly, we also deal with the time period t differently in the encryption algorithm for our scheme $\mathbf{\Pi}_1$. As a result, we no longer need $\mathbf{T}_{\mathbf{A}}$ to answer all the queries made by the adversary $\mathcal{A}$ in the security proof.

The items in $\mathbf{\Pi}_0, \mathbf{\Pi}_1$ related to the above changes are boxed in Figure 1. Besides, we describe the underlined items in $\mathbf{\Pi}_0, \mathbf{\Pi}_1$ as follows (the items in $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ are not marked since there is no simplification). For the scheme $\mathbf{\Pi}_0$ in Figure 1, the vector $\mathbf{f}_{\mathsf{ID},k}$ in $\mathsf{SK}_{\mathsf{ID}}$, the vector $\mathbf{f}_{\mathsf{ID},\mathsf{t},k}$ in $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ and the vector $\mathbf{d}_{\mathsf{ID},\mathsf{t}}$ in $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$, satisfy the condition $\mathbf{f}_{\mathsf{ID},\mathsf{t},k} = \mathbf{d}_{\mathsf{ID},\mathsf{t}} + [\mathbf{f}_{\mathsf{ID},k} \| \mathbf{0}_{m\times 1}] \in \mathbb{Z}^{(\ell+2)m}$ for $k \in [\ell+1, L]$, where $[\cdot\|\cdot]$ denotes vertical concatenation of vectors, and $\ell = |\mathsf{ID}|$. Actually, as a preparation for achieving DKER, Katsumata et al. [11] also presented an RHIBE scheme without DKER, where the decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ does not contain the item $\mathbf{g}_{\mathsf{ID},\mathsf{t}}$. Following this scheme without DKER, they introduced these vectors $\mathbf{f}_{\mathsf{ID},k}, \mathbf{f}_{\mathsf{ID}_{[i]},\mathsf{t},k}$ to avoid a trivial attack. For simplicity, one can imagine that if there is no $\mathbf{g}_{\mathsf{ID},\mathsf{t}}$ in $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ for our scheme $\mathbf{\Pi}_1$ in Figure 1, then the private $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ is totally contained in the public $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$, which is obviously insecure. However, for the construction of RHIBE with the DKER setting, it can be proved that the item $\mathbf{g}_{\mathsf{ID},\mathsf{t}}$ itself is sufficient to guarantee the security. Therefore, one no longer needs the items $(\mathbf{f}_{\mathsf{ID},k})_{k\in[\ell+1,L]}$ in $\mathsf{SK}_{\mathsf{ID}}$, or part of the items $(\mathbf{f}_{\mathsf{ID}_{[i]},\mathsf{t},k})_{(i,k)\in[\ell]\times[\ell+1,L]}$ in $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$. Then in the public parameters PP we can also use only one vector $\mathbf{u}$, instead of $(\mathbf{u}_k)_{k\in[L]}$, and finally our scheme $\mathbf{\Pi}_1$ is obtained as a simplification of $\mathbf{\Pi}_0$, shown as in Figure 1.

As for our second RHIBE scheme $\mathbf{\Pi}_2$, we follow the idea of our $\mathbf{\Pi}_1$, and adopt the technique for lattice basis delegation in fixed dimension introduced in [2]. Therefore, the sizes of items are much smaller than $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_1$. For example, the ciphertext CT under an identity ID with $\ell = |\mathsf{ID}|$, is a vector in $\mathbb{Z}_q^{(2\ell+1)m+1}$ for our $\mathbf{\Pi}_2$. While in $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_1$, CT is a vector in $\mathbb{Z}_q^{(\frac{1}{2}\ell^2+\frac{7}{2}\ell+2)m+1}$. Moreover, as Figure 1 shows, the items in $\mathsf{SK}_{\mathsf{ID}}$, $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$, $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ for $\mathbf{\Pi}_2$ do not depend on $\ell = |\mathsf{ID}|$. They are either short matrices in $\mathbb{Z}^{m\times m}$, or short vectors in $\mathbb{Z}^m$ or $\mathbb{Z}^{2m}$. Unlike only one matrix $\mathbf{T}_{\mathbf{A}}$ in $\mathsf{SK}_{\mathsf{KGC}}$ for $\mathbf{\Pi}_1$, we emphasize that the master secret key $\mathsf{SK}_{\mathsf{KGC}}$ in our scheme $\mathbf{\Pi}_2$ contains two trapdoor bases $\mathbf{T}_{\mathbf{A}}, \mathbf{T}_{\mathbf{B}}$. This comes from the different technique introduced in [2]. Following this, two trapdoor bases are necessary even for the construction of RIBE (not RHIBE) without DKER.

Organization. The rest of this paper is organized as follows. Section 2 reviews some background on lattices, the definitions for RHIBE re-formalized in [11], and the complete subtree method. Then in Section 3, we provide our first RHIBE scheme $\mathbf{\Pi}_1$, together with its analysis. The construction and the security proof for our second RHIBE scheme $\mathbf{\Pi}_2$, are presented in Section 4. Finally, the conclusion is given in Section 5.

2 Preliminaries

Notations. The acronym PPT stands for "probabilistic polynomial-time". We say that a function $\epsilon : \mathbb{N} \to \mathbb{R}$ is negligible, if for sufficient large $\lambda \in \mathbb{N}$, $|\epsilon(\lambda)|$ is smaller than the reciprocal of any polynomial in λ. The notation $\mathsf{negl}(\lambda)$ is used to denote a negligible function $\epsilon(\lambda)$. Besides, an event is said to happen with overwhelming probability if it happens with probability at least $1 - \mathsf{negl}(\lambda)$. The statistical distance of two random variables X and Y over a discrete domain $\varOmega$ is defined as $\varDelta(X;Y) := \frac{1}{2}\sum_{s\in\varOmega} |\Pr[X=s] - \Pr[Y=s]|$. If $X(\lambda)$ and $Y(\lambda)$ are ensembles of random variables, we say that X and Y are statistically close if $d(\lambda) := \varDelta(X(\lambda);Y(\lambda))$ is equal to $\mathsf{negl}(\lambda)$. For a distribution χ, we often write $x \hookleftarrow \chi$ to indicate that we sample x from χ. For a finite set $\varOmega$, the notation $x \overset{\$}{\leftarrow} \varOmega$ means that x is chosen uniformly at random from $\varOmega$. We treat vectors in their column form. For a vector $\mathbf{x} \in \mathbb{Z}^n$, denote $\|\mathbf{x}\|$ as the Euclidean norm of $\mathbf{x}$. For a matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$, denote $\|\mathbf{A}\|$ as the Euclidean norm of the longest column in $\mathbf{A}$, and denote $\|\mathbf{A}\|_{\mathrm{GS}}$ as $\|\mathbf{A}_{\mathrm{GS}}\|$, where $\mathbf{A}_{\mathrm{GS}}$ is the Gram-Schmidt orthogonalization of $\mathbf{A}$.

2.1 Background on Lattices

Integer Lattices. A (full-rank) integer lattice Λ of dimension m is defined as the set $\big\{\sum_{i\in[m]} x_i\mathbf{b}_i \mid x_i \in \mathbb{Z}\big\}$, where $\mathbf{B} := \{\mathbf{b}_1, \cdots, \mathbf{b}_m\}$ are m linearly independent vectors in $\mathbb{Z}^m$. Here $\mathbf{B}$ is called the basis of the lattice Λ. Let n, m and $q \geqslant 2$ be positive integers. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define the m-dimensional lattice $\Lambda_q^{\perp}(\mathbf{A}) := \big\{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{Ax} = \mathbf{0} \bmod q\big\}$. For any $\mathbf{u}$ in the image of $\mathbf{A}$, define the coset $\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \big\{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{Ax} = \mathbf{u} \bmod q\big\}$.

Discrete Gaussians over Lattices. Let Λ be a lattice in $\mathbb{Z}^m$. For any parameter $\sigma \in \mathbb{R}_{>0}$, define $\rho_\sigma(\mathbf{x}) := \exp(-\pi\|\mathbf{x}\|^2/\sigma^2)$ for $\mathbf{x} \in \mathbb{Z}^m$, and $\rho_\sigma(\Lambda) := \sum_{\mathbf{x}\in\Lambda}\rho_\sigma(\mathbf{x})$. The discrete Gaussian distribution over Λ with parameter σ is $\mathcal{D}_{\Lambda,\sigma}(\mathbf{y}) := \rho_\sigma(\mathbf{y})/\rho_\sigma(\Lambda)$, for $\mathbf{y} \in \Lambda$. Some properties are shown as follows.

Lemma 1. *([16]) For $\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{u} \in \mathbb{Z}_q^n$ with $q \geqslant 2, m > n$, let $\mathbf{T_A}$ be a basis for $\Lambda_q^{\perp}(\mathbf{A})$ and $\sigma \geqslant \|\mathbf{T_A}\|_{\mathrm{GS}} \cdot \omega(\sqrt{\log m})$, then* $\Pr[\mathbf{x} \hookleftarrow \mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma} : \|\mathbf{x}\| > \sigma\sqrt{m}] \leqslant \mathsf{negl}(n)$.

Lemma 2. *([9]) Suppose that $n, m, q \in \mathbb{Z}_{>0}$, $\sigma \in \mathbb{R}_{>0}$, with q a prime, $m \geqslant 2n\log q$ and $\sigma \geqslant \omega(\sqrt{\log n})$. Then for $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{e} \hookleftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$, the distribution of $\mathbf{u} := \mathbf{Ae} \pmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. Furthermore, for a fixed vector $\mathbf{u} \in \mathbb{Z}_q^n$ and a matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, the conditional distribution of $\mathbf{e} \hookleftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$ given $\mathbf{Ae} = \mathbf{u} \pmod q$ is $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$ with overwhelming probability.*

In addition, as in [2], we set $\sigma_{\mathbf{R}} := \sqrt{n\log q} \cdot \omega(\sqrt{\log m})$, and let $\mathcal{D}_{m\times m}$ denote the distribution on matrices in $\mathbb{Z}^{m\times m}$ defined as $(\mathcal{D}_{\mathbb{Z}^m,\sigma_{\mathbf{R}}})^m$ conditioned on the resulting matrix being $\mathbb{Z}_q$-invertible.

Algorithms about Lattices. Let us briefly review some algorithms which are useful for lattice-based cryptography. For these algorithms introduced below, we simply assume that $n, m, m_0, q \in \mathbb{Z}_{>0}$ with $q \geqslant 3$ a prime and $m = \Omega(n\log q)$. Besides, we note that according to [15], there exists a fixed full rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$, called the gadget matrix, such that the lattice $\Lambda_q^{\perp}(\mathbf{G})$ has a publicly known basis $\mathbf{T_G} \in \mathbb{Z}^{m\times m}$ with $\|\mathbf{T_G}\|_{\mathrm{GS}} \leqslant \sqrt{5}$.

$\mathsf{TrapGen}(1^n, 1^m, q) \to (\mathbf{A}, \mathbf{T_A})$ ([3,4,15]): On input n, m, q, output a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, such that $\mathbf{A}$ is distributed statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ and $\|\mathbf{T_A}\|_{\mathrm{GS}} \leqslant O(\sqrt{n\log q})$ with overwhelming probability in n.

$\mathsf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{u}, \sigma) \to \mathbf{e}$ ([9]): On input a full rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter $\sigma \geqslant \|\mathbf{T_A}\|_{\mathrm{GS}} \cdot \omega(\sqrt{\log m})$, output a vector $\mathbf{e} \in \mathbb{Z}^m$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$.

$\mathsf{SampleLeft}(\mathbf{A}, \mathbf{M}, \mathbf{T_A}, \mathbf{u}, \sigma) \to \mathbf{e}$ ([1,7]): On input a full rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times m_0}$, a basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter $\sigma \geqslant \|\mathbf{T_A}\|_{\mathrm{GS}} \cdot \omega(\sqrt{\log(m + m_0)})$, output a vector $\mathbf{e} \in \mathbb{Z}^{m+m_0}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}([\mathbf{A}|\mathbf{M}]),\sigma}$.

$\mathsf{SampleRight}(\mathbf{A}, \mathbf{H}\cdot\mathbf{G}, \mathbf{R}, \mathbf{T_G}, \mathbf{u}, \sigma) \to \mathbf{e}$ ([1,15]): On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a matrix of the form $\mathbf{H}\cdot\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ (where $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ is full rank and $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is the gadget matrix [15]), a uniform random matrix $\mathbf{R} \xleftarrow{\$} \{-1, 1\}^{m\times m}$, a basis $\mathbf{T_G}$ of $\Lambda_q^{\perp}(\mathbf{G})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter $\sigma \geqslant \|\mathbf{T_G}\|_{\mathrm{GS}} \cdot \sqrt{m} \cdot \omega(\sqrt{\log m})$, output a vector $\mathbf{e} \in \mathbb{Z}^{2m}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}([\mathbf{A}|\mathbf{AR}+\mathbf{HG}]),\sigma}$.

$\mathsf{RandBasis}(\mathbf{T}, \sigma) \to \mathbf{T}'$ ([7]): On input a basis $\mathbf{T}$ of an m-dimensional lattice $\Lambda_q^{\perp}(\mathbf{A})$ and a Gaussian parameter $\sigma \geqslant \|\mathbf{T}\|_{\mathrm{GS}} \cdot \omega(\sqrt{\log m})$, output a new basis $\mathbf{T}'$ of $\Lambda_q^{\perp}(\mathbf{A})$ such that $\mathbf{T}'$ is distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^{\perp}(\mathbf{A}), \sigma)$ introduced below, and $\|\mathbf{T}'\|_{\mathrm{GS}} \leqslant \sigma\sqrt{m}$ holds with overwhelming probability.

The distribution $\mathcal{D}_{Basis}(\Lambda_q^\perp(\mathbf{A}),\sigma)$ used above can be briefly described as follows. Let $\mathcal{O}(\Lambda_q^\perp(\mathbf{A}),\sigma)$ be an algorithm that generates samples from the distribution $\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}),\sigma}$, and set m as the dimension of $\Lambda_q^\perp(\mathbf{A})$. For $i=1,2,\cdots,m$, run $\mathbf{v}\leftarrow\mathcal{O}(\Lambda_q^\perp(\mathbf{A}),\sigma)$ repeatedly until $\mathbf{v}$ is linearly independent of $\{\mathbf{v}_1,\cdots,\mathbf{v}_{i-1}\}$, and then set $\mathbf{v}_i\leftarrow\mathbf{v}$. After that, convert the set of vectors $\{\mathbf{v}_1,\cdots,\mathbf{v}_m\}$ to a basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$ using Lemma 7.1 of [14] (and using some canonical basis of $\Lambda_q^\perp(\mathbf{A})$). The distribution of this $\mathbf{T}_\mathbf{A}$ is then denoted as $\mathcal{D}_{Basis}(\Lambda_q^\perp(\mathbf{A}),\sigma)$. Actually, in the process of $\mathsf{RandBasis}(\mathbf{T},\sigma)\rightarrow\mathbf{T}'$, the input basis $\mathbf{T}$ is only used to run the algorithm $\mathsf{SamplePre}(\mathbf{A},\mathbf{T},\mathbf{0},\sigma)$, instead of the above algorithm $\mathcal{O}(\Lambda_q^\perp(\mathbf{A}),\sigma)$. Thus up to a negligible statistical distance, the distribution of the output basis $\mathbf{T}'$ does not depend on $\mathbf{T}$.

Using the distribution $\mathcal{D}_{Basis}(\Lambda_q^\perp(\mathbf{A}),\sigma)$ introduced above, we are able to describe the following algorithms for generating a random basis of some lattice.

$\mathsf{SampleBasisLeft}(\mathbf{A},\mathbf{M},\mathbf{T}_\mathbf{A},\sigma)\rightarrow\mathbf{T}_{[\mathbf{A}|\mathbf{M}]}$ ([1,7]): On input a full rank matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$, a matrix $\mathbf{M}\in\mathbb{Z}_q^{n\times m_0}$, a basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$, and a Gaussian parameter $\sigma\geqslant\|\mathbf{T}_\mathbf{A}\|_{\mathrm{GS}}\cdot\omega(\sqrt{\log(m+m_0)})$, output a basis $\mathbf{T}_{[\mathbf{A}|\mathbf{M}]}\in\mathbb{Z}^{(m+m_0)\times(m+m_0)}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp([\mathbf{A}\mid\mathbf{M}]),\sigma)$.

$\mathsf{SampleBasisRight}(\mathbf{A},\mathbf{H}\cdot\mathbf{G},\mathbf{R},\mathbf{T}_\mathbf{G},\sigma)\rightarrow\mathbf{T}_{[\mathbf{A}|\mathbf{AR}+\mathbf{HG}]}$ ([1,15]): On input a matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$, a matrix of the form $\mathbf{H}\cdot\mathbf{G}\in\mathbb{Z}_q^{n\times m}$ (where $\mathbf{H}\in\mathbb{Z}_q^{n\times n}$ is full rank and $\mathbf{G}\in\mathbb{Z}_q^{n\times m}$ is the gadget matrix [15]), a uniform random matrix $\mathbf{R}\xleftarrow{\$}\{-1,1\}^{m\times m}$, a basis $\mathbf{T}_\mathbf{G}$ of $\Lambda_q^\perp(\mathbf{G})$, and a Gaussian parameter $\sigma\geqslant\|\mathbf{T}_\mathbf{G}\|_{\mathrm{GS}}\cdot\sqrt{m}\cdot\omega(\sqrt{\log m})$, output a basis $\mathbf{T}_{[\mathbf{A}|\mathbf{AR}+\mathbf{HG}]}\in\mathbb{Z}^{2m\times 2m}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp([\mathbf{A}\mid\mathbf{AR}+\mathbf{HG}]),\sigma)$.

$\mathsf{BasisDel}(\mathbf{A},\mathbf{R},\mathbf{T}_\mathbf{A},\sigma)\rightarrow\mathbf{T}_{(\mathbf{AR}^{-1})}$ ([2]): On input a full rank matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$, a $\mathbb{Z}_q$-invertible matrix $\mathbf{R}\in\mathbb{Z}^{m\times m}$ sampled from $\mathcal{D}_{m\times m}$, a basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$, and a Gaussian parameter $\sigma\geqslant\|\mathbf{T}_\mathbf{A}\|_{\mathrm{GS}}\cdot\sqrt{nm\log q}\cdot\omega(\log^2 m)$, output a basis $\mathbf{T}_{(\mathbf{AR}^{-1})}\in\mathbb{Z}^{m\times m}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp(\mathbf{AR}^{-1}),\sigma)$.

$\mathsf{SampleRwithBasis}(\mathbf{A},\sigma)\rightarrow(\mathbf{R},\mathbf{T}_{(\mathbf{AR}^{-1})})$ ([2,7]): On input a full rank matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$, and a Gaussian parameter $\sigma\geqslant\sqrt{n\log q}\cdot\omega(\sqrt{\log m})$, output a $\mathbb{Z}_q$-invertible matrix $\mathbf{R}\in\mathbb{Z}^{m\times m}$ sampled from a distribution statistically close to $\mathcal{D}_{m\times m}$, and a basis $\mathbf{T}_{(\mathbf{AR}^{-1})}\in\mathbb{Z}^{m\times m}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp(\mathbf{AR}^{-1}),\sigma)$.

Recall that the distribution $\mathcal{D}_{m\times m}$ used above has already been defined below Lemma 2. Besides, the algorithm $\mathsf{SampleRwithBasis}$ described above is actually a combination of the original algorithm $\mathsf{SampleRwithBasis}$ in [2] and the algorithm $\mathsf{RandBasis}$ in [7]. We directly describe this modified $\mathsf{SampleRwithBasis}$ just for convenience in the future proof of security.

Hardness Assumption. The learning with errors (LWE) problem, first introduced by Regev [18], plays a central role in lattice-based cryptography. The security of our schemes will rely on the following LWE assumption.

Assumption 1 (LWE) *Suppose that $n,m,q\in\mathbb{Z}_{>0}$, $\alpha\in(0,1)$ with q a prime satisfy $\alpha q>2\sqrt{n}$. For a PPT algorithm $\mathcal{A}$, the advantage for the learning with errors problem $\mathsf{LWE}_{n,m,q,\mathcal{D}_{\mathbb{Z}^m,\alpha q}}$ of $\mathcal{A}$ is defined as $|\Pr[\mathcal{A}(\mathbf{A},\mathbf{A}^\top\mathbf{s}+\mathbf{x})=1]-\Pr[\mathcal{A}(\mathbf{A},\mathbf{v})=1]|$, where $\mathbf{A}\xleftarrow{\$}\mathbb{Z}_q^{n\times m},\mathbf{s}\xleftarrow{\$}\mathbb{Z}_q^n,\mathbf{x}\hookleftarrow\mathcal{D}_{\mathbb{Z}^m,\alpha q},\mathbf{v}\xleftarrow{\$}\mathbb{Z}_q^m$. We say that the LWE assumption holds if the above advantage is negligible for all PPT $\mathcal{A}$.*

2.2 Revocable Hierarchical Identity-Based Encryption

We briefly review the syntax, correctness and security definition for RHIBE, which are reformalized in [11]. First of all, let us introduce some notations as follows.

Recall that the hierarchical identity space in RHIBE is denoted by $\mathcal{ID}_\mathsf{H}=(\mathcal{ID})^{\leqslant L}=\bigcup_{i\in[L]}(\mathcal{ID})^i$, where $\mathcal{ID}$ is the element identity space, and L is the maximum depth of the hierarchy. The KGC is the unique level-0 identity, and an identity $\mathsf{ID}\in\mathcal{ID}_\mathsf{H}$ at level $\ell\in[L]$ is expressed as a length-ℓ vector $\mathsf{ID}=(\mathsf{id}_1,\cdots,\mathsf{id}_\ell)\in(\mathcal{ID})^\ell$. For $k\in[\ell]$, we set $\mathsf{ID}_{[k]}:=$

$(\mathsf{id}_1, \cdots, \mathsf{id}_k)$ as the length-k prefix of ID, and define $\mathsf{prefix}(\mathsf{ID}) := \{\mathsf{ID}_{[1]},\ \mathsf{ID}_{[2]},\ \cdots, \mathsf{ID}_{[\ell]} = \mathsf{ID}\}$. Besides, we let $\mathsf{pa}(\mathsf{ID}) := \mathsf{ID}_{[\ell-1]}$ if $\ell \geqslant 2$, and $\mathsf{pa}(\mathsf{ID}) := \mathsf{KGC}$ if $\ell = 1$. Here $\mathsf{pa}(\mathsf{ID})$ is called the parent of ID. We use $\mathsf{ID}\|\mathcal{ID}$ to denote the subset of $(\mathcal{ID})^{\ell+1}$ which contains all the members that have $\mathsf{ID} \in (\mathcal{ID})^{\ell}$ as its parent. When $\mathsf{ID} = \mathsf{KGC}$ (i.e. $\ell = 0$), the notation $\mathsf{ID}\|\mathcal{ID}$ just denotes $\mathcal{ID}$.

Next, we introduce the notation RL_t $\left(\subseteq (\mathcal{ID})^{\leqslant L}\right)$ to denote the revocation list on the time period t. If $\mathsf{ID} \in \mathsf{RL}_\mathsf{t}$, then implicitly we assume $\mathsf{ID}' \in \mathsf{RL}_\mathsf{t}$ also holds, where ID' is any descendant of ID. Besides, it is required that $\mathsf{RL}_{\mathsf{t}_1} \subseteq \mathsf{RL}_{\mathsf{t}_2}$ for $\mathsf{t}_1 < \mathsf{t}_2$. We set $\mathsf{RL}_{\mathsf{ID},\mathsf{t}} := \mathsf{RL}_\mathsf{t} \cap (\mathsf{ID}\|\mathcal{ID})$ as the revocation list managed by the identity ID on the time period t. Following these notations, when we write "$\mathsf{ID} \in \mathsf{RL}_\mathsf{t}$", it means that user ID has been revoked on the time period t. For any $\mathsf{ID}' \in \mathsf{prefix}(\mathsf{ID})$ and any $\mathsf{t}' \leqslant \mathsf{t}$, we have $\mathsf{ID}' \in \mathsf{RL}_{\mathsf{pa}(\mathsf{ID}'),\mathsf{t}'} \Rightarrow \mathsf{ID} \in \mathsf{RL}_\mathsf{t}$. When we write "$\mathsf{ID} \notin \mathsf{RL}_\mathsf{t}$", it means that user ID is not revoked on the time period t. We have $\mathsf{ID} \notin \mathsf{RL}_\mathsf{t} \Leftrightarrow \mathsf{ID}' \notin \mathsf{RL}_{\mathsf{pa}(\mathsf{ID}'),\mathsf{t}}, \forall\ \mathsf{ID}' \in \mathsf{prefix}(\mathsf{ID})$.

Syntax. As re-formalized in [11], an RHIBE scheme $\mathbf{\Pi}$ consists of the following six algorithms **Setup**, **Encrypt**, **GenSK**, **KeyUp**, **GenDK**, **Decrypt**. Here the "revoke" algorithm is not explicitly introduced, since it is a simple operation of appending revoked users into a revocation list.

Setup$(1^\lambda, 1^L) \to (\mathsf{PP}, \mathsf{SK}_\mathsf{KGC})$: This is the setup algorithm run by the KGC. On input a security parameter λ and the maximum depth of the hierarchy L, it outputs public parameters PP and the KGC's secret key SK_KGC.

Encrypt$(\mathsf{PP}, \mathsf{ID}, \mathsf{t}, \mathsf{M}) \to \mathsf{CT}$: This is the encryption algorithm run by a sender. On input public parameters PP, an identity ID, a time period t, and a plaintext M, it outputs a ciphertext CT.

GenSK$(\mathsf{PP}, \mathsf{SK}_{\mathsf{pa}(\mathsf{ID})}, \mathsf{ID}) \to (\mathsf{SK}_\mathsf{ID}, \mathsf{SK}'_{\mathsf{pa}(\mathsf{ID})})$: This is the secret key generation algorithm run by $\mathsf{pa}(\mathsf{ID})$, the parent user of ID. On input public parameters PP, the parent user's secret key $\mathsf{SK}_{\mathsf{pa}(\mathsf{ID})}$, and the identity ID, it outputs a secret key SK_ID for ID along with the parent user's "updated" secret key $\mathsf{SK}'_{\mathsf{pa}(\mathsf{ID})}$.

KeyUp$(\mathsf{PP}, \mathsf{t}, \mathsf{SK}_\mathsf{ID}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}}, \mathsf{KU}_{\mathsf{pa}(\mathsf{ID}),\mathsf{t}}) \to (\mathsf{KU}_{\mathsf{ID},\mathsf{t}}, \mathsf{SK}'_\mathsf{ID})$: This is the key update generation algorithm run by the user ID. On input public parameters PP, a time period t, a secret key SK_ID, a revocation list $\mathsf{RL}_{\mathsf{ID},\mathsf{t}}$, and the parent user's key update $\mathsf{KU}_{\mathsf{pa}(\mathsf{ID}),\mathsf{t}}$, it outputs a key update $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ along with the "updated" secret key SK'_ID. (In the special case $\mathsf{ID} = \mathsf{KGC}$, since $\mathsf{KU}_{\mathsf{pa}(\mathsf{KGC}),\mathsf{t}}$ is not needed, we just define $\mathsf{KU}_{\mathsf{pa}(\mathsf{KGC}),\mathsf{t}} := \perp$ for all $\mathsf{t} \in \mathcal{T}$.)

GenDK$(\mathsf{PP}, \mathsf{SK}_\mathsf{ID}, \mathsf{KU}_{\mathsf{pa}(\mathsf{ID}),\mathsf{t}}) \to \mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ or $\perp$: This is the decryption key generation algorithm run by the user ID. On input public parameters PP, a secret key SK_ID, and the parent user's key update $\mathsf{KU}_{\mathsf{pa}(\mathsf{ID}),\mathsf{t}}$, it outputs a decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$, or the special "invalid" symbol $\perp$ which indicates that ID has been revoked.

Decrypt$(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{t}}, \mathsf{CT}) \to \mathsf{M}$: This is the decryption algorithm run by the user ID. On input public parameters PP, a decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$, and a ciphertext CT, it outputs the decrypted plaintext M.

Correctness. The correctness requirement for an RHIBE scheme $\mathbf{\Pi}$ states that, for all $\lambda, L \in \mathbb{Z}_{>0}$, $\ell \in [L]$, $\mathsf{ID} \in (\mathcal{ID})^{\ell}$, $\mathsf{t} \in \mathcal{T}$, $\mathsf{M} \in \mathcal{M}$, $\mathsf{RL}_\mathsf{t} \subseteq (\mathcal{ID})^{\leqslant L}$, if $\mathsf{ID} \notin \mathsf{RL}_\mathsf{t}$, and all parties follow the above prescribed algorithms **Setup**, **GenSK**, **KeyUp**, **GenDK**, **Encrypt** to generate $\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{t}}, \mathsf{CT}$, then **Decrypt**$(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{t}}, \mathsf{CT}) = \mathsf{M}$.

Security Definition. Let $\mathbf{\Pi} = $ (**Setup**, **Encrypt**, **GenSK**, **KeyUp**, **GenDK**, **Decrypt**) be an RHIBE scheme. We first consider the selective-identity security, which is defined via the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

At the beginning, $\mathcal{A}$ sends the challenge identity/time period pair $(\mathsf{ID}^*, \mathsf{t}^*) \in (\mathcal{ID})^{\leqslant L} \times \mathcal{T}$ to $\mathcal{C}$. After that, $\mathcal{C}$ runs $(\mathsf{PP}, \mathsf{SK}_\mathsf{KGC}) \leftarrow$ **Setup**$(1^\lambda, 1^L)$, and prepares a list SKList that initially contains $(\mathsf{KGC}, \mathsf{SK}_\mathsf{KGC})$. During the game, whenever a new secret key is generated or an existing secret key is updated for some identity $\mathsf{ID} \in \{\mathsf{KGC}\} \cup (\mathcal{ID})^{\leqslant L}$, the challenger $\mathcal{C}$ will store or update the identity/secret key pairs $(\mathsf{ID}, \mathsf{SK}_{\mathsf{ID}})$ in SKList, and we do not explicitly mention this addition/update. The global counter t_{cu}, which denotes the "current time period", is initialized with 1. Then $\mathcal{C}$ executes $(\mathsf{KU}_{\mathsf{KGC},1}, \mathsf{SK}'_{\mathsf{KGC}}) \leftarrow \mathbf{KeyUp}(\mathsf{PP}, \mathsf{t}_{\mathsf{cu}} = 1, \mathsf{SK}_{\mathsf{KGC}}, \mathsf{RL}_{\mathsf{KGC},1} = \emptyset, \perp)$ for $\mathsf{t}_{\mathsf{cu}} = 1$, and gives $\mathsf{PP}, \mathsf{KU}_{\mathsf{KGC},1}$ to $\mathcal{A}$.

From this point on, $\mathcal{A}$ may adaptively make the following five types of queries to $\mathcal{C}$.

Secret Key Generation Query: Upon a query $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L}$ from $\mathcal{A}$, the challenger $\mathcal{C}$ checks whether the condition $(\mathsf{ID}, *) \notin \mathsf{SKList}$, $(\mathsf{pa}(\mathsf{ID}), \mathsf{SK}_{\mathsf{pa}(\mathsf{ID})}) \in \mathsf{SKList}$ is satisfied. If not, $\mathcal{C}$ just returns $\perp$. Otherwise, $\mathcal{C}$ executes $(\mathsf{SK}_{\mathsf{ID}}, \mathsf{SK}'_{\mathsf{pa}(\mathsf{ID})}) \leftarrow \mathbf{GenSK}(\mathsf{PP}, \mathsf{SK}_{\mathsf{pa}(\mathsf{ID})}, \mathsf{ID})$. Furthermore, if $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L-1}$, then $\mathcal{C}$ executes $(\mathsf{KU}_{\mathsf{ID},\mathsf{t}_{\mathsf{cu}}}, \mathsf{SK}'_{\mathsf{ID}}) \leftarrow \mathbf{KeyUp}(\mathsf{PP}, \mathsf{t}_{\mathsf{cu}}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}_{\mathsf{cu}}} = \emptyset, \mathsf{KU}_{\mathsf{pa}(\mathsf{ID}),\mathsf{t}_{\mathsf{cu}}})$, and returns $\mathsf{KU}_{\mathsf{ID},\mathsf{t}_{\mathsf{cu}}}$ to $\mathcal{A}$.

Secret Key Reveal Query: Upon a query $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L}$ from $\mathcal{A}$, the challenger $\mathcal{C}$ checks whether the following condition is satisfied.
- If $\mathsf{t}_{\mathsf{cu}} \geqslant \mathsf{t}^*$ and $\mathsf{ID} \in \mathsf{prefix}(\mathsf{ID}^*)$, then $\mathsf{ID} \in \mathsf{RL}_{\mathsf{t}^*}$.

If not, $\mathcal{C}$ just returns $\perp$. Otherwise, $\mathcal{C}$ finds $\mathsf{SK}_{\mathsf{ID}}$ from SKList, and returns it to $\mathcal{A}$.

Revoke & Key Update Query: Upon a query $\mathsf{RL} \subseteq (\mathcal{ID})^{\leqslant L}$ from $\mathcal{A}$, the challenger $\mathcal{C}$ checks whether the following conditions are satisfied simultaneously.
- $\mathsf{RL}_{\mathsf{t}_{\mathsf{cu}}} \subseteq \mathsf{RL}$.
- For $\mathsf{ID}, \mathsf{ID}' \in (\mathcal{ID})^{\leqslant L}$ with $\mathsf{ID}' \in \mathsf{prefix}(\mathsf{ID})$, if $\mathsf{ID}' \in \mathsf{RL}$, then $\mathsf{ID} \in \mathsf{RL}$.
- If $\mathsf{t}_{\mathsf{cu}} = \mathsf{t}^* - 1$, and $\mathsf{SK}_{\mathsf{ID}}$ for some $\mathsf{ID} \in \mathsf{prefix}(\mathsf{ID}^*)$ has been revealed by the secret key reveal query, then $\mathsf{ID} \in \mathsf{RL}$.

If not, $\mathcal{C}$ just returns $\perp$. Otherwise, $\mathcal{C}$ increments the current time period by $\mathsf{t}_{\mathsf{cu}} \leftarrow \mathsf{t}_{\mathsf{cu}} + 1$, and then sets $\mathsf{RL}_{\mathsf{t}_{\mathsf{cu}}} \leftarrow \mathsf{RL}$. Next, for all $\mathsf{ID} \in \{\mathsf{KGC}\} \cup (\mathcal{ID})^{\leqslant L-1}$ with $(\mathsf{ID}, *) \in \mathsf{SKList}$, $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}_{\mathsf{cu}}}$ in the breadth-first order in the identity hierarchy, $\mathcal{C}$ set $\mathsf{RL}_{\mathsf{ID},\mathsf{t}_{\mathsf{cu}}} \leftarrow \mathsf{RL}_{\mathsf{t}_{\mathsf{cu}}} \cap (\mathsf{ID} \| \mathcal{ID})$, and run $(\mathsf{KU}_{\mathsf{ID},\mathsf{t}_{\mathsf{cu}}}, \mathsf{SK}'_{\mathsf{ID}}) \leftarrow \mathbf{KeyUp}(\mathsf{PP}, \mathsf{t}_{\mathsf{cu}}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}_{\mathsf{cu}}}, \mathsf{KU}_{\mathsf{pa}(\mathsf{ID}),\mathsf{t}_{\mathsf{cu}}})$. Finally, $\mathcal{C}$ returns all these generated key updates $\{\mathsf{KU}_{\mathsf{ID},\mathsf{t}_{\mathsf{cu}}}\}$ to $\mathcal{A}$.

Decryption Key Reveal Query: Upon a query $(\mathsf{ID},\mathsf{t}) \in (\mathcal{ID})^{\leqslant L} \times \mathcal{T}$ from $\mathcal{A}$, the challenger $\mathcal{C}$ checks whether the following condition is satisfied.
- $\mathsf{t} \leqslant \mathsf{t}_{\mathsf{cu}}$, $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, $(\mathsf{ID},\mathsf{t}) \neq (\mathsf{ID}^*, \mathsf{t}^*)$.

If not, $\mathcal{C}$ just returns $\perp$. Otherwise, $\mathcal{C}$ finds $\mathsf{SK}_{\mathsf{ID}}$ from SKList, runs $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} \leftarrow \mathbf{GenDK}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{pa}(\mathsf{ID}),\mathsf{t}})$, and returns $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ to $\mathcal{A}$.

Challenge Query: $\mathcal{A}$ is allowed to make this query only once. Upon a query $(\mathsf{M}_0, \mathsf{M}_1)$ with $|\mathsf{M}_0| = |\mathsf{M}_1|$ from $\mathcal{A}$, the challenger $\mathcal{C}$ picks the challenge bit $b \xleftarrow{\$} \{0, 1\}$, runs $\mathsf{CT}^* \leftarrow \mathbf{Encrypt}(\mathsf{PP}, \mathsf{ID}^*, \mathsf{t}^*, \mathsf{M}_b)$, and returns the challenge ciphertext CT^* to $\mathcal{A}$.

At some point, $\mathcal{A}$ outputs $b' \in \{0, 1\}$ as the guess for b and terminates.

The above completes the description of the game. In this game, $\mathcal{A}$'s selective-identity security advantage is defined by $\mathsf{Adv}^{\mathsf{RHIBE\text{-}sel}}_{\mathbf{\Pi},L,\mathcal{A}}(\lambda) := 2 \cdot |\Pr[b' = b] - 1/2|$, where λ is the security parameter. We say that an RHIBE scheme $\mathbf{\Pi}$ with depth L satisfies the selective-identity security, if the advantage $\mathsf{Adv}^{\mathsf{RHIBE\text{-}sel}}_{\mathbf{\Pi},L,\mathcal{A}}(\lambda)$ is negligible for any PPT adversary $\mathcal{A}$.

The game for the adaptive-identity security, is defined in the same way as the above game, except that the adversary $\mathcal{A}$ chooses the challenge identity/time period pair $(\mathsf{ID}^*, \mathsf{t}^*) \in (\mathcal{ID})^{\leqslant L} \times \mathcal{T}$ not at the beginning of the game, but at the time when $\mathcal{A}$ makes the challenge query. Formally, the challenge query is defined differently as follows.

Challenge Query: $\mathcal{A}$ is allowed to make this query only once. The query $(\mathsf{ID}^*, \mathsf{t}^*, \mathsf{M}_0, \mathsf{M}_1)$ from $\mathcal{A}$ must satisfy the following conditions simultaneously.
- $|\mathsf{M}_0| = |\mathsf{M}_1|$.
- If $\mathsf{t}_{\mathsf{cu}} \geqslant \mathsf{t}^*$, and $\mathsf{SK}_{\mathsf{ID}}$ for some $\mathsf{ID} \in \mathsf{prefix}(\mathsf{ID}^*)$ has been revealed by the secret key reveal query, then $\mathsf{ID} \in \mathsf{RL}_{\mathsf{t}^*}$.
- If $\mathsf{t}_{\mathsf{cu}} \geqslant \mathsf{t}^*$, then $\mathcal{A}$ has not submitted $(\mathsf{ID}^*, \mathsf{t}^*)$ as a decryption key reveal query.

After receiving this query $(\mathsf{ID}^*, \mathsf{t}^*, \mathsf{M}_0, \mathsf{M}_1)$, $\mathcal{C}$ picks the challenge bit $b \xleftarrow{\$} \{0, 1\}$, runs $\mathsf{CT}^* \leftarrow \mathbf{Encrypt}(\mathsf{PP}, \mathsf{ID}^*, \mathsf{t}^*, \mathsf{M}_b)$, and returns the challenge ciphertext CT^* to $\mathcal{A}$.

Besides, in the other queries, the conditions related to $\mathsf{ID}^*, \mathsf{t}^*$ are naturally omitted before $\mathcal{A}$ makes the above challenge query. Recall that at last $\mathcal{A}$ will output $b' \in \{0,1\}$ as the guess for b. The adaptive-identity security advantage is then defined by $\mathsf{Adv}^{\mathsf{RHIBE\text{-}ad}}_{\mathbf{\Pi},L,\mathcal{A}}(\lambda) := 2 \cdot |\Pr[b' = b] - 1/2|$ for this modified game. Similarly, we say that an RHIBE scheme $\mathbf{\Pi}$ with depth L satisfies the adaptive-identity security, if the advantage $\mathsf{Adv}^{\mathsf{RHIBE\text{-}ad}}_{\mathbf{\Pi},L,\mathcal{A}}(\lambda)$ is negligible for any PPT adversary $\mathcal{A}$.

2.3 The Complete Subtree Method

Similar to the works in [5,8], the RHIBE scheme $\mathbf{\Pi}_0$ in [11], and our schemes $\mathbf{\Pi}_1$, $\mathbf{\Pi}_2$ constructed in this paper, all need the complete subtree (CS) method of Naor et al. [17] to achieve the revocation mechanism.

Shown as in Figure 1, every identity ID, including the KGC, keeps a binary tree $\mathsf{BT}_{\mathsf{ID}}$ in its secret key $\mathsf{SK}_{\mathsf{ID}}$. Actually, each member that has ID as its parent, will be randomly assigned to a leaf node of $\mathsf{BT}_{\mathsf{ID}}$. For a leaf node η, we use $\mathsf{Path}(\mathsf{BT}_{\mathsf{ID}}, \eta)$ to denote the set of nodes on the path from η to the root in $\mathsf{BT}_{\mathsf{ID}}$ (both η and the root inclusive). For a non-leaf node θ, let θ_l, θ_r denote the left and right child of θ, respectively. Besides, recall that $\mathsf{RL}_{\mathsf{ID},\mathsf{t}}$ is the revocation list managed by the identity ID on the time period t. Then the algorithm KUNode, which takes $\mathsf{BT}_{\mathsf{ID}}$ and $\mathsf{RL}_{\mathsf{ID},\mathsf{t}}$ as input, can be described as follows: (1) $X, Y \leftarrow \emptyset$; (2) for each $\mathsf{ID}' \in \mathsf{RL}_{\mathsf{ID},\mathsf{t}}$, add $\mathsf{Path}(\mathsf{BT}_{\mathsf{ID}}, \eta_{\mathsf{ID}'})$ to X, where $\eta_{\mathsf{ID}'}$ denotes the leaf node to which ID' is assigned; (3) for each node $\theta \in X$, add θ_l to Y if $\theta_l \notin X$, and add θ_r to Y if $\theta_r \notin X$; (4) if $\mathsf{RL}_{\mathsf{ID},\mathsf{t}} = \emptyset$, add the root node of $\mathsf{BT}_{\mathsf{ID}}$ to Y; (5) return Y as the output of $\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})$.

Let us focus on the decryption key generation algorithm $\mathbf{GenDK}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}'}, \mathsf{KU}_{\mathsf{ID},\mathsf{t}})$ run by the user ID' with $\mathsf{pa}(\mathsf{ID}') = \mathsf{ID}$. Here the secret key $\mathsf{SK}_{\mathsf{ID}'}$ contains the set of nodes $\mathsf{P} := \mathsf{Path}(\mathsf{BT}_{\mathsf{ID}}, \eta_{\mathsf{ID}'})$. While the key update $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ contains the set of nodes $\mathsf{K} := \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})$. If $\mathsf{ID}' \notin \mathsf{RL}_{\mathsf{ID},\mathsf{t}}$, we have $\mathsf{P} \cap \mathsf{K} = \{\theta^*\}$, which contains exactly one node θ^*. Then ID' is able to generate its decryption key $\mathsf{DK}_{\mathsf{ID}',\mathsf{t}}$, using some item related to θ^*. If $\mathsf{ID}' \in \mathsf{RL}_{\mathsf{ID},\mathsf{t}}$, we have $\mathsf{P} \cap \mathsf{K} = \emptyset$, from which ID' can never obtain $\mathsf{DK}_{\mathsf{ID}',\mathsf{t}}$. This is the general way to achieve the revocation mechanism from the CS method.

3 RHIBE Scheme in the Standard Model

In this section, we describe our first RHIBE scheme $\mathbf{\Pi}_1$ in Section 3.1, and then present its selective-identity security in Section 3.2. As a preparation, we need to explain our treatment of some spaces such as $\mathcal{T}, \mathcal{ID}, \mathcal{ID}_{\mathsf{H}} = (\mathcal{ID})^{\leqslant L}$, and introduce an encoding with full-rank differences used in the scheme $\mathbf{\Pi}_1$.

Treatment of Spaces. The element identity space $\mathcal{ID}$ is treated as a subset of $\mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}$, namely, $\mathcal{ID} \subset \mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}$. We need to define a function $f : \mathcal{ID} \to \widetilde{\mathcal{ID}}$ such that $f(\mathsf{id}_1) \neq f(\mathsf{id}_2)$ for $\mathsf{id}_1 \neq \mathsf{id}_2$. Here $\widetilde{\mathcal{ID}}$ is a new space satisfying $\widetilde{\mathcal{ID}} \subset \mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}$ and $\mathcal{ID} \cap \widetilde{\mathcal{ID}} = \emptyset$. For simplicity, we just define

$$\mathcal{ID} := \{1\} \times \mathbb{Z}_q^{n-1},\ \widetilde{\mathcal{ID}} := \{2\} \times \mathbb{Z}_q^{n-1} \quad \text{and} \quad f(1\|\mathbf{v}) := 2\|\mathbf{v} \ \text{ for } \ \mathbf{v} \in \mathbb{Z}_q^{n-1}.$$

The time period space $\mathcal{T} = \{1, 2, \cdots, \mathsf{t}_{\max}\}$ is encoded into the set $\mathbb{Z}_q^{n-1}$. Here we note that one can also choose disjoint $\mathcal{ID}, \widetilde{\mathcal{ID}} \subset \mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}$ such that $|\mathcal{ID}| = |\widetilde{\mathcal{ID}}| = \frac{1}{2}(q^n - 1)$, and set $\mathcal{T}$ as a subset of $\mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}$ with $|\mathcal{T}| = \lfloor \frac{1}{L}(q^n - 1) \rfloor$. Besides, let us deal with the hierarchical identity space $\mathcal{ID}_{\mathsf{H}} = (\mathcal{ID})^{\leqslant L} = \bigcup_{i \in [L]} (\mathcal{ID})^i$. Define $\mathcal{F} : (\mathcal{ID})^{\leqslant L} \to \bigcup_{i \in [0, L-1]} (\mathcal{ID})^i \times \widetilde{\mathcal{ID}}$ as $\mathcal{F}(\mathsf{ID}) := (\mathsf{id}_1, \cdots, \mathsf{id}_{\ell-1}, f(\mathsf{id}_\ell))$ for $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_{\ell-1}, \mathsf{id}_\ell)$. Thus for $|\mathsf{ID}| = \ell \geqslant 2$, we have $\mathsf{ID} \neq \mathcal{F}(\mathsf{ID})$, $\mathsf{ID}_{[\ell-1]} = [\mathcal{F}(\mathsf{ID})]_{[\ell-1]}$. For simplicity, let us set $\widetilde{\mathsf{id}} := f(\mathsf{id})$, $\widetilde{\mathsf{ID}} := \mathcal{F}(\mathsf{ID})$, and use $\widetilde{\mathsf{ID}_{[i]}}$ to denote $\mathcal{F}(\mathsf{ID}_{[i]})$.

Encoding with Full-Rank Differences. We use the standard map H defined in [1] to encode vectors as matrices. The function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is actually an encoding with full-rank differences for a prime q. Namely, the matrix $H(\mathsf{ch}_1) - H(\mathsf{ch}_2)$ is full rank for any two distinct $\mathsf{ch}_1, \mathsf{ch}_2 \in \mathbb{Z}_q^n$, and H is computable in polynomial time in $n \log q$. One can refer to [1] for the explicit construction of the map H. Finally, for $\mathsf{CH} = (\mathsf{ch}_1, \mathsf{ch}_2, \cdots, \mathsf{ch}_\ell) \in \left(\mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}\right)^{\leqslant L}$ and $i \in [L],\ \mathsf{t} \in \mathbb{Z}_q^{n-1}$, we define the following functions:

- $\mathbf{E}(\mathsf{CH}) := [\mathbf{C}_1 + H(\mathsf{ch}_1)\mathbf{G} \mid \mathbf{C}_2 + H(\mathsf{ch}_2)\mathbf{G} \mid \cdots \mid \mathbf{C}_\ell + H(\mathsf{ch}_\ell)\mathbf{G}] \in \mathbb{Z}_q^{n\times \ell m}$,
- $\mathbf{F}(i, \mathsf{t}) := \mathbf{C}_{L+1} + H(i\|\mathsf{t})\mathbf{G} \in \mathbb{Z}_q^{n\times m}$.

Here $(\mathbf{C}_i)_{i\in[L+1]}$ are uniformly random matrices in $\mathbb{Z}_q^{n\times m}$ chosen in the setup algorithm of the scheme $\mathbf{\Pi}_1$ and $\mathbf{G}$ is the gadget matrix [15]. In addition, we can treat $i\|\mathsf{t}$ as a vector in $\mathbb{Z}_q^n$, since $L < q$ obviously holds due to the parameters selection given later.

3.1 Construction

Due to our new treatment of the identity spaces and the time period space, we can obtain a much simple RHIBE scheme $\mathbf{\Pi}_1$ in the standard model, which is described as follows. Here we let $\alpha, \alpha', (\sigma_\ell)_{\ell\in[0,L]}$ be positive reals denoting Gaussian parameters, and set N as the maximum number of children each parent manages. These parameters, together with positive integers n, m and a prime q, are all implicitly determined by the security parameter λ, and in particular we set $n(\lambda) := \lambda$.

Setup$(1^n, 1^L) \to (\mathsf{PP}, \mathsf{SK}_{\mathsf{KGC}})$:

Taking the security parameter n and the maximum depth of the hierarchy L as input, it performs the following steps.

1. Run $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. Select $\mathbf{C}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ for $i \in [L+1]$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. Create a binary tree $\mathsf{BT}_{\mathsf{KGC}}$ with N leaf nodes, which denote N children users.
4. Output $\mathsf{PP} := \Big(\mathbf{A},\ (\mathbf{C}_i)_{i\in[L+1]},\ \mathbf{u}\Big)$, $\mathsf{SK}_{\mathsf{KGC}} := \Big(\mathsf{BT}_{\mathsf{KGC}},\ \mathbf{T}_\mathbf{A}\Big)$.

Here recall that $(\mathbf{C}_i)_{i\in[L+1]}$ define the functions $\mathbf{E}(\cdot)$ and $\mathbf{F}(\cdot)$ introduced before.

Encrypt$(\mathsf{PP}, \mathsf{ID}, \mathsf{t}, \mathsf{M}) \to \mathsf{CT}$:

For $\mathsf{M} \in \{0,1\},\ |\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Select $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$ for $i \in [\ell] \cup \{L+1\}$. Then sample $x \hookleftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, $\mathbf{x}_i \hookleftarrow \mathcal{D}_{\mathbb{Z}^{(i+2)m},\alpha' q}$ for $i \in [\ell]$, and $\mathbf{x}_{L+1} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{(\ell+2)m},\alpha' q}$.
2. Set
$$\begin{cases} c_0 := \mathbf{u}^\top(\mathbf{s}_1 + \mathbf{s}_2 + \cdots + \mathbf{s}_\ell) + \mathbf{u}^\top \mathbf{s}_{L+1} + x + \mathsf{M}\lfloor \frac{q}{2} \rfloor, \\ \mathbf{c}_i := [\mathbf{A} \mid \mathbf{E}(\widetilde{\mathsf{ID}_{[i]}}) \mid \mathbf{F}(i,\mathsf{t})]^\top \mathbf{s}_i + \mathbf{x}_i \quad \text{for} \quad i \in [\ell], \\ \mathbf{c}_{L+1} := [\mathbf{A} \mid \mathbf{E}(\mathsf{ID}) \mid \mathbf{F}(\ell,\mathsf{t})]^\top \mathbf{s}_{L+1} + \mathbf{x}_{L+1}. \end{cases}$$
3. Output $\mathsf{CT} := \Big(c_0,\ (\mathbf{c}_i)_{i\in[\ell]},\ \mathbf{c}_{L+1}\Big) \in \mathbb{Z}_q \times (\mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{4m} \times \cdots \times \mathbb{Z}_q^{(\ell+2)m}) \times \mathbb{Z}_q^{(\ell+2)m}$.

GenSK$(\mathsf{PP}, \mathsf{SK}_{\mathsf{pa(ID)}}, \mathsf{ID}) \to (\mathsf{SK}_{\mathsf{ID}}, \mathsf{SK}'_{\mathsf{pa(ID)}})$:

For $|\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Randomly pick an unassigned leaf node η_{ID} from $\mathsf{BT}_{\mathsf{pa(ID)}}$ and store ID in node η_{ID}. Then select $\mathbf{u}_{\mathsf{pa(ID)},\theta} \xleftarrow{\$} \mathbb{Z}_q^n$ for node $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$, if $\mathbf{u}_{\mathsf{pa(ID)},\theta}$ is undefined. Here $\mathsf{pa(ID)}$ updates $\mathsf{SK}_{\mathsf{pa(ID)}}$ to $\mathsf{SK}'_{\mathsf{pa(ID)}}$ by storing new defined $\mathbf{u}_{\mathsf{pa(ID)},\theta}$ in $\theta \in \mathsf{BT}_{\mathsf{pa(ID)}}$.
2. Run $\mathbf{e}_{\mathsf{ID},\theta} \leftarrow \mathsf{SampleLeft}([\mathbf{A} \mid \mathbf{E}(\mathsf{pa(ID)})], \mathbf{C}_\ell + H(\widetilde{\mathsf{id}_\ell})\mathbf{G}, \mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{pa(ID)})]}, \mathbf{u}_{\mathsf{pa(ID)},\theta}, \sigma_{\ell-1})$ for $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$. Here $\mathbf{e}_{\mathsf{ID},\theta} \in \mathbb{Z}^{(\ell+1)m}$ satisfies $[\mathbf{A} \mid \mathbf{E}(\widetilde{\mathsf{ID}})]\mathbf{e}_{\mathsf{ID},\theta} = \mathbf{u}_{\mathsf{pa(ID)},\theta}$.
3. Run $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID})]} \leftarrow \mathsf{SampleBasisLeft}([\mathbf{A} \mid \mathbf{E}(\mathsf{pa(ID)})], \mathbf{C}_\ell + H(\mathsf{id}_\ell)\mathbf{G}, \mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{pa(ID)})]}, \sigma_{\ell-1})$.
4. Create a new binary tree $\mathsf{BT}_{\mathsf{ID}}$ with N leaf nodes.

5. Output $\mathsf{SK}_{\mathsf{ID}} := \Big(\mathsf{BT}_{\mathsf{ID}},\ (\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})},\ \mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID})]}\Big),\ \mathsf{SK}'_{\mathsf{pa(ID)}}$.

KeyUp$(\mathsf{PP}, \mathsf{t}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}) \to (\mathsf{KU}_{\mathsf{ID},\mathsf{t}}, \mathsf{SK}'_{\mathsf{ID}})$:

For $|\mathsf{ID}| = \ell \in [0, L-1]$, it performs the following steps.

1. Select $\mathbf{u}_{\mathsf{ID},\theta} \xleftarrow{\$} \mathbb{Z}_q^n$ for node $\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})$, if $\mathbf{u}_{\mathsf{ID},\theta}$ is undefined. Here ID may update $\mathsf{SK}_{\mathsf{ID}}$ to $\mathsf{SK}'_{\mathsf{ID}}$ by storing new defined $\mathbf{u}_{\mathsf{ID},\theta}$ in $\theta \in \mathsf{BT}_{\mathsf{ID}}$.
2. Run $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} \leftarrow \mathsf{SampleLeft}([\mathbf{A} \mid \mathbf{E}(\mathsf{ID})],\ \mathbf{F}(\ell+1, \mathsf{t}),\ \mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID})]},\ \mathbf{u} - \mathbf{u}_{\mathsf{ID},\theta},\ \sigma_\ell)$ for $\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})$. Here $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} \in \mathbb{Z}^{(\ell+2)m}$ satisfies $[\mathbf{A} \mid \mathbf{E}(\mathsf{ID}) \mid \mathbf{F}(\ell+1, \mathsf{t})]\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} = \mathbf{u} - \mathbf{u}_{\mathsf{ID},\theta}$.
3. If $\ell \geqslant 1$, run $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} \leftarrow$ **GenDK**$(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}})$, where **GenDK**$(\cdot)$ is defined below. Then extract $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell]}$ from $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$.
4. Output $\mathsf{KU}_{\mathsf{ID},\mathsf{t}} := \Big((\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})},\ (\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell]}\Big),\ \mathsf{SK}'_{\mathsf{ID}}$.

GenDK$(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}) \to \mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ or $\perp$:

For $|\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Extract $\mathsf{P} := \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$ in $\mathsf{SK}_{\mathsf{ID}}$, and $\mathsf{K} := \mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa(ID)}}, \mathsf{RL}_{\mathsf{pa(ID)},\mathsf{t}})$ in $\mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}$. If $\mathsf{P} \cap \mathsf{K} = \emptyset$, output $\perp$. Otherwise, for the unique node $\theta^* \in \mathsf{P} \cap \mathsf{K}$, extract $\mathbf{e}_{\mathsf{ID},\theta^*}, \mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \in \mathbb{Z}^{(\ell+1)m}$ in $\mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}$, respectively. Parse them as $\mathbf{e}_{\mathsf{ID},\theta^*} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta^*}]$, $\mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*}]$, where $\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta^*}, \mathbf{e}^{\mathsf{L}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \in \mathbb{Z}^{\ell m}$ and $\mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta^*}, \mathbf{e}^{\mathsf{R}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \in \mathbb{Z}^m$. Then set $\mathbf{d}_{\mathsf{ID},\mathsf{t}} := [\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta^*} + \mathbf{e}^{\mathsf{L}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*}] \in \mathbb{Z}^{(\ell+2)m}$.
2. If $\ell \geqslant 2$, extract $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell-1]}$ from $\mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}$.
3. Run $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \leftarrow \mathsf{SampleLeft}([\mathbf{A} \mid \mathbf{E}(\mathsf{ID})],\ \mathbf{F}(\ell, \mathsf{t}),\ \mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID})]},\ \mathbf{u},\ \sigma_\ell)$. Here $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \in \mathbb{Z}^{(\ell+2)m}$ satisfies $[\mathbf{A} \mid \mathbf{E}(\mathsf{ID}) \mid \mathbf{F}(\ell, \mathsf{t})]\mathbf{g}_{\mathsf{ID},\mathsf{t}} = \mathbf{u}$.
4. Output $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} := \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$.

Decrypt$(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{t}}, \mathsf{CT}) \to \mathsf{M}$:

For $|\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Compute $c' := c_0 - \sum_{i=1}^{\ell} \mathbf{d}^\top_{\mathsf{ID}_{[i]},\mathsf{t}} \mathbf{c}_i - \mathbf{g}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{c}_{L+1} \in \mathbb{Z}_q$. Treat c' as an integer in $[q] \subset \mathbb{Z}$.
2. Output $\mathsf{M} := 1$ if $|c' - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$, and output $\mathsf{M} := 0$ otherwise.

Correctness. Assume that ID has the depth $|\mathsf{ID}| = \ell \in [L]$. If $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, then one can obtain $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} = \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$. Recall that $\mathbf{d}_{\mathsf{ID},\mathsf{t}} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta^*} + \mathbf{e}^{\mathsf{L}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*}] \in \mathbb{Z}^{(\ell+2)m}$, where $\theta^* \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}}) \cap \mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa(ID)}}, \mathsf{RL}_{\mathsf{pa(ID)},\mathsf{t}})$. According to

$$\begin{aligned} &[\mathbf{A} \mid \mathbf{E}(\widetilde{\mathsf{ID}})]\mathbf{e}_{\mathsf{ID},\theta^*} = \mathbf{u}_{\mathsf{pa(ID)},\theta^*}, && [\mathbf{A} \mid \mathbf{E}(\mathsf{pa(ID)}) \mid \mathbf{F}(\ell, \mathsf{t})]\mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} = \mathbf{u} - \mathbf{u}_{\mathsf{pa(ID)},\theta^*}, \\ &\mathbf{e}_{\mathsf{ID},\theta^*} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{ID},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{ID},\theta^*}], && \mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} = [\mathbf{e}^{\mathsf{L}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \| \mathbf{e}^{\mathsf{R}}_{\mathsf{pa(ID)},\mathsf{t},\theta^*}], \end{aligned}$$

one can obtain $[\mathbf{A} \mid \mathbf{E}(\widetilde{\mathsf{ID}}) \mid \mathbf{F}(\ell, \mathsf{t})]\mathbf{d}_{\mathsf{ID},\mathsf{t}} = \mathbf{u}$, $\mathbf{d}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{c}_\ell = \mathbf{u}^\top \mathbf{s}_\ell + \mathbf{d}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{x}_\ell$. Similarly, for $i \in [\ell-1]$ we also have $[\mathbf{A} \mid \mathbf{E}(\widetilde{\mathsf{ID}_{[i]}}) \mid \mathbf{F}(i, \mathsf{t})]\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}} = \mathbf{u}$, $\mathbf{d}^\top_{\mathsf{ID}_{[i]},\mathsf{t}} \mathbf{c}_i = \mathbf{u}^\top \mathbf{s}_i + \mathbf{d}^\top_{\mathsf{ID}_{[i]},\mathsf{t}} \mathbf{x}_i$. Besides, the vector $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \in \mathbb{Z}^{(\ell+2)m}$ satisfies $[\mathbf{A} \mid \mathbf{E}(\mathsf{ID}) \mid \mathbf{F}(\ell, \mathsf{t})]\mathbf{g}_{\mathsf{ID},\mathsf{t}} = \mathbf{u}$, $\mathbf{g}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{c}_{L+1} = \mathbf{u}^\top \mathbf{s}_{L+1} + \mathbf{g}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{x}_{L+1}$. From the above, we can compute

$$\begin{aligned} c' &= \mathbf{u}^\top (\textstyle\sum_{i=1}^{\ell} \mathbf{s}_i) + \mathbf{u}^\top \mathbf{s}_{L+1} + x + \mathsf{M}\lfloor \frac{q}{2} \rfloor - \sum_{i=1}^{\ell} \mathbf{d}^\top_{\mathsf{ID}_{[i]},\mathsf{t}} \mathbf{c}_i - \mathbf{g}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{c}_{L+1} \\ &= \mathsf{M}\lfloor \tfrac{q}{2} \rfloor + (x - \textstyle\sum_{i=1}^{\ell} \mathbf{d}^\top_{\mathsf{ID}_{[i]},\mathsf{t}} \mathbf{x}_i - \mathbf{g}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{x}_{L+1}). \end{aligned}$$

Set $z := x - \sum_{i=1}^{\ell} \mathbf{d}^\top_{\mathsf{ID}_{[i]},\mathsf{t}} \mathbf{x}_i - \mathbf{g}^\top_{\mathsf{ID},\mathsf{t}} \mathbf{x}_{L+1}$ as the noise. Then according to the triangle inequality, the Cauchy-Schwarz inequality, and Lemma 1, the noise z can be bounded as follows with overwhelming probability:

$$\begin{aligned} \left|z\right| &\leqslant \left|x\right| + \textstyle\sum_{i=1}^{\ell} \left\|\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}\right\| \cdot \left\|\mathbf{x}_i\right\| + \left\|\mathbf{g}_{\mathsf{ID},\mathsf{t}}\right\| \cdot \left\|\mathbf{x}_{L+1}\right\| \\ &\leqslant \alpha q + \textstyle\sum_{i=1}^{\ell} 2 \cdot \sigma_{i-1}\sqrt{(i+2)m} \cdot \alpha' q \sqrt{(i+2)m} + \sigma_\ell \sqrt{(\ell+2)m} \cdot \alpha' q \sqrt{(\ell+2)m} \\ &= \alpha q + [\textstyle\sum_{i=1}^{\ell} 2(i+2)\sigma_{i-1} + (\ell+2)\sigma_\ell] m\alpha' q \\ &\leqslant \alpha q + [2L(L+2) + (L+2)]\sigma_L m \alpha' q \\ &= O(\alpha q + L^2 \sigma_L m \alpha' q). \end{aligned}$$

As a conclusion, if $O(\alpha q + L^2\sigma_L m\alpha' q) < q/5$, we know that $\left|z\right|$ is upper bounded by $q/5$ with overwhelming probability, and thus our RHIBE scheme $\mathbf{\Pi}_1$ only has negligible decryption error.

Parameters. The analysis for parameters selection is similar to that in [11]. We must consider the condition $O(\alpha q + L^2\sigma_L m\alpha' q) < q/5$ for the correctness requirement, and the condition $q > 2\sqrt{n}/\alpha$ for the hardness assumption of $\mathsf{LWE}_{n,m+1,q,\mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}}$. Besides, we also need to make sure that algorithms such as $\mathsf{SampleBasisLeft}$ et al. can operate in the construction, and algorithms such as $\mathsf{SampleBasisRight}$ et al. can work in the security proof. Finally, we set the parameters used for our RHIBE scheme $\mathbf{\Pi}_1$ as follows:

$$\begin{aligned} &m = 6n^{1+\delta} = O(Ln\log n), && \alpha = [L^{\frac{5}{2}} m^{\frac{1}{2}L+2} \omega(\log^{\frac{1}{2}L+\frac{1}{2}} n)]^{-1}, && \alpha' = O((Lm)^{\frac{1}{2}})\alpha, \\ &q = L^{\frac{5}{2}} m^{\frac{1}{2}L+\frac{5}{2}} \omega(\log^{\frac{1}{2}L+\frac{1}{2}} n), && \sigma_\ell = m^{\frac{1}{2}\ell+\frac{1}{2}} \omega(\log^{\frac{1}{2}\ell+\frac{1}{2}} n) \ \text{ for } \ \ell \in [0, L], \end{aligned}$$

and round up m to the nearest larger integer, and q to the nearest larger prime. Here we choose δ such that $n^\delta > \lceil \log q \rceil = O(L\log n)$.

3.2 Security

Theorem 1. *The RHIBE scheme $\mathbf{\Pi}_1$ satisfies the selective-identity security, assuming the hardness of the problem $\mathsf{LWE}_{n,m+1,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.*

Let $\mathsf{ID}^* = (\mathsf{id}_1^*, \cdots, \mathsf{id}_{\ell^*}^*)$, t^* be the challenge identity and time period with $\ell^* := |\mathsf{ID}^*|$. Then the attack strategies taken by $\mathcal{A}$ can be divided into the following two types, which consist of $\ell^* + 1$ strategies in total.

- **Type-I**: $\mathcal{A}$ issues secret key reveal queries on at least one $\mathsf{ID} \in \mathsf{prefix}(\mathsf{ID}^*)$.
  - Further divided into **Type-I-i^*** ($i^* \in [\ell^*]$):
    $\mathcal{A}$ issues a secret key reveal query on $\mathsf{ID}^*_{[i^*]}$ but not on any $\mathsf{ID} \in \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]})$.
- **Type-II**: $\mathcal{A}$ does not issue secret key reveal queries on any $\mathsf{ID} \in \mathsf{prefix}(\mathsf{ID}^*)$.

We follow the framework in [12] (the full version of [11]) to show the security proof. According to the "strategy-dividing lemma" introduced in [11,12], the following Lemma 3 and Lemma 4 are sufficient for the proof of Theorem 1. Thus it remains to prove these two lemmas. Since they are similar to the proof in [12], the proofs of Lemma 3 and Lemma 4 are presented in Appendix A and Appendix B, respectively.

Lemma 3. *Suppose that a PPT adversary $\mathcal{A}$ follows the* **Type-I-i^*** *strategy for some $i^* \in [\ell^*]$. Then its advantage is negligible, assuming the hardness of the problem $\mathsf{LWE}_{n,m+1,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.*

Lemma 4. *Suppose that a PPT adversary $\mathcal{A}$ follows the* **Type-II** *strategy. Then its advantage is negligible, assuming the hardness of the problem $\mathsf{LWE}_{n,m+1,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.*

4 RHIBE Scheme in the Random Oracle Model

In this section, we describe our second RHIBE scheme $\mathbf{\Pi}_2$ in Section 4.1, and then provide the proof of its adaptive-identity security in Section 4.2. As a preparation, we need to explain our treatment of some spaces such as $\mathcal{T}, \mathcal{ID}, \mathcal{ID}_{\mathsf{H}} = (\mathcal{ID})^{\leqslant L}$, and introduce two random oracles used in the scheme $\mathbf{\Pi}_2$.

Treatment of Spaces. The time period space $\mathcal{T}$, the element identity space $\mathcal{ID}$, and the space $\widetilde{\mathcal{ID}}$ are all treated as subsets of $\{0,1,2\}^{\omega}$, such that $\mathcal{T} \cap \mathcal{ID} = \mathcal{T} \cap \widetilde{\mathcal{ID}} = \mathcal{ID} \cap \widetilde{\mathcal{ID}} = \emptyset$. Here ω is an integer determined by the security parameter. Similarly, we also need to define a function $f : \mathcal{ID} \to \widetilde{\mathcal{ID}}$ satisfying $f(\mathsf{id}_1) \neq f(\mathsf{id}_2)$ for $\mathsf{id}_1 \neq \mathsf{id}_2$. For simplicity, we just define

$$\mathcal{T} := \{0\} \times \{0,1,2\}^{\omega-1}, \ \mathcal{ID} := \{1\} \times \{0,1,2\}^{\omega-1}, \ \widetilde{\mathcal{ID}} := \{2\} \times \{0,1,2\}^{\omega-1},$$
$$\text{and} \ \ f(1\|\mathsf{ch}) := 2\|\mathsf{ch} \ \ \text{for} \ \ \mathsf{ch} \in \{0,1,2\}^{\omega-1}.$$

Note that here one can also choose $\mathcal{T}, \mathcal{ID}, \widetilde{\mathcal{ID}}$ as pairwise disjoint subsets of $\{0,1\}^*$. Next, let us deal with the hierarchical identity space $\mathcal{ID}_{\mathsf{H}} = (\mathcal{ID})^{\leqslant L} = \bigcup_{i \in [L]} (\mathcal{ID})^i$. We still define $\mathcal{F} : (\mathcal{ID})^{\leqslant L} \to \bigcup_{i \in [0, L-1]} (\mathcal{ID})^i \times \widetilde{\mathcal{ID}}$ as $\mathcal{F}(\mathsf{ID}) := (\mathsf{id}_1, \cdots, \mathsf{id}_{\ell-1}, f(\mathsf{id}_\ell))$ for $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_{\ell-1}, \mathsf{id}_\ell)$. Similarly, for $|\mathsf{ID}| = \ell \geqslant 2$, we have $\mathsf{ID} \neq \mathcal{F}(\mathsf{ID})$, $\mathsf{ID}_{[\ell-1]} = [\mathcal{F}(\mathsf{ID})]_{[\ell-1]}$. For simplicity, we still set $\widetilde{\mathsf{id}} := f(\mathsf{id})$, $\widetilde{\mathsf{ID}} := \mathcal{F}(\mathsf{ID})$, and use $\widetilde{\mathsf{ID}_{[i]}}$ to denote $\mathcal{F}(\mathsf{ID}_{[i]})$. In addition, for KGC and $\mathsf{ID} = (\mathsf{id}_1, \cdots, \mathsf{id}_\ell) \in (\{0,1,2\}^{\omega})^{\ell}$, we define the notations $\mathsf{KGC}\|\mathsf{t} := \mathsf{t}$ and $\mathsf{ID}\|\mathsf{t} := (\mathsf{id}_1, \cdots, \mathsf{id}_\ell, \mathsf{t}) \in (\{0,1,2\}^{\omega})^{\ell+1}$, and thus $(\mathsf{ID}\|\mathsf{t})_{[\ell]} = \mathsf{ID}$.

Random Oracles. We define two random oracles $\mathbf{H}_1, \mathbf{H}_2$ as follows:

- $\mathbf{H}_1 : (\{0,1,2\}^{\omega})^{\leqslant L+1} \to \mathbb{Z}_q^{m \times m}, \ \ \mathsf{CH} \mapsto \mathbf{H}_1(\mathsf{CH}) \sim \mathcal{D}_{m \times m}$,
- $\mathbf{H}_2 : (\{0,1,2\}^{\omega})^{\leqslant L} \to \mathbb{Z}_q^{m \times m}, \ \ \mathsf{CH}' \mapsto \mathbf{H}_2(\mathsf{CH}') \sim \mathcal{D}_{m \times m}$.

Here the outputs of $\mathbf{H}_1, \mathbf{H}_2$ are both distributed as $\mathcal{D}_{m \times m}$, which is defined below Lemma 2 in Section 2.1. Furthermore, for $\mathsf{CH} \in (\{0,1,2\}^{\omega})^{\leqslant L+1}$ with $\ell = |\mathsf{CH}|$, and $\mathsf{CH}' \in (\{0,1,2\}^{\omega})^{\leqslant L}$ with $\ell' = |\mathsf{CH}'|$, we define the following functions:

- $\mathbf{P}_1(\mathsf{CH}) := [\mathbf{H}_1(\mathsf{CH}_{[\ell]})\mathbf{H}_1(\mathsf{CH}_{[\ell-1]}) \cdots \mathbf{H}_1(\mathsf{CH}_{[1]})]^{-1} \in \mathbb{Z}_q^{m \times m}$,
- $\mathbf{P}_2(\mathsf{CH}') := [\mathbf{H}_2(\mathsf{CH}'_{[\ell']})\mathbf{H}_2(\mathsf{CH}'_{[\ell'-1]}) \cdots \mathbf{H}_2(\mathsf{CH}'_{[1]})]^{-1} \in \mathbb{Z}_q^{m \times m}$.

Therefore, after setting $\mathbf{P}_1(\mathsf{CH}_{[0]}), \mathbf{P}_2(\mathsf{CH}'_{[0]})$ as the identity matrix $\mathbf{I}_{m \times m}$, we have $\mathbf{P}_1(\mathsf{CH}_{[j]}) = \mathbf{P}_1(\mathsf{CH}_{[j-1]}) \cdot [\mathbf{H}_1(\mathsf{CH}_{[j]})]^{-1}$ for $j \in [\ell]$, and $\mathbf{P}_2(\mathsf{CH}'_{[j']}) = \mathbf{P}_2(\mathsf{CH}'_{[j'-1]}) \cdot [\mathbf{H}_2(\mathsf{CH}'_{[j']})]^{-1}$ for $j' \in [\ell']$.

4.1 Construction

In the following, we describe the construction of our RHIBE scheme $\mathbf{\Pi}_2$ in the random oracle model. Note that in this scheme the KGC's secret key $\mathsf{SK}_{\mathsf{KGC}}$ contains two trapdoor bases. Similar to Section 3.1, here we let $\alpha, (\sigma_\ell)_{\ell \in [0,L]}$ be positive reals denoting Gaussian parameters, and set N as the maximum number of children each parent manages. These parameters, together with positive integers n, m and a prime q, are all implicitly determined by the security parameter λ, and in particular we set $n(\lambda) := \lambda$. Besides, in the scheme $\mathbf{\Pi}_2$ we set $\tau_\ell := \sigma_\ell \sqrt{m} \cdot \omega(\sqrt{\log m})$ for $\ell \in [0, L]$ to make the algorithm $\mathsf{SamplePre}$ work.

Setup$(1^n, 1^L) \to (\mathsf{PP}, \mathsf{SK}_{\mathsf{KGC}})$:

Taking the security parameter n and the maximum depth of the hierarchy L as input, it performs the following steps.

1. Run $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, and $(\mathbf{B}, \mathbf{T_B}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. Select $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$.

3. Create a binary tree $\mathsf{BT}_{\mathsf{KGC}}$ with N leaf nodes, which denote N children users.
4. Output $\mathsf{PP} := \Big(\mathbf{A},\ \mathbf{B},\ \mathbf{u}\Big)$, $\mathsf{SK}_{\mathsf{KGC}} := \Big(\mathsf{BT}_{\mathsf{KGC}},\ \mathbf{T}_{\mathbf{A}},\ \mathbf{T}_{\mathbf{B}}\Big)$.

Encrypt$(\mathsf{PP}, \mathsf{ID}, \mathsf{t}, \mathsf{M}) \to \mathsf{CT}$:

For $\mathsf{M} \in \{0, 1\}$, $|\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Select $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$ for $i \in [\ell] \cup \{L+1\}$. Then sample $x \hookleftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, $\mathbf{x}_{i,j} \hookleftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$ for $(i, j) \in [\ell] \times [2]$, and $\mathbf{x}_{L+1} \hookleftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$.
2. Define $\mathsf{ID}_{[0]}\|\mathsf{t} := \mathsf{t}$, and then set
$$\begin{cases} c_0 := \mathbf{u}^\top(\mathbf{s}_1 + \mathbf{s}_2 + \cdots + \mathbf{s}_\ell) + \mathbf{u}^\top \mathbf{s}_{L+1} + x + \mathsf{M}\lfloor \frac{q}{2} \rfloor, \\ \mathbf{c}_{i,1} := [\mathbf{A} \cdot \mathbf{P}_1(\widetilde{\mathsf{ID}_{[i]}})]^\top \mathbf{s}_i + \mathbf{x}_{i,1} \quad \text{for} \quad i \in [\ell], \\ \mathbf{c}_{i,2} := [\mathbf{B} \cdot \mathbf{P}_2(\mathsf{ID}_{[i-1]}\|\mathsf{t})]^\top \mathbf{s}_i + \mathbf{x}_{i,2} \quad \text{for} \quad i \in [\ell], \\ \mathbf{c}_{L+1} := [\mathbf{A} \cdot \mathbf{P}_1(\mathsf{ID}\|\mathsf{t})]^\top \mathbf{s}_{L+1} + \mathbf{x}_{L+1}. \end{cases}$$
3. Output $\mathsf{CT} := \Big(c_0,\ (\mathbf{c}_{i,1}, \mathbf{c}_{i,2})_{i\in[\ell]},\ \mathbf{c}_{L+1}\Big) \in \mathbb{Z}_q \times (\mathbb{Z}_q^m \times \mathbb{Z}_q^m)^\ell \times \mathbb{Z}_q^m$.

GenSK$(\mathsf{PP}, \mathsf{SK}_{\mathsf{pa(ID)}}, \mathsf{ID}) \to (\mathsf{SK}_{\mathsf{ID}}, \mathsf{SK}'_{\mathsf{pa(ID)}})$:

For $|\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Randomly pick an unassigned leaf node η_{ID} from $\mathsf{BT}_{\mathsf{pa(ID)}}$ and store ID in node η_{ID}. Then select $\mathbf{u}_{\mathsf{pa(ID)},\theta} \xleftarrow{\$} \mathbb{Z}_q^n$ for node $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$, if $\mathbf{u}_{\mathsf{pa(ID)},\theta}$ is undefined. Here $\mathsf{pa(ID)}$ updates $\mathsf{SK}_{\mathsf{pa(ID)}}$ to $\mathsf{SK}'_{\mathsf{pa(ID)}}$ by storing new defined $\mathbf{u}_{\mathsf{pa(ID)},\theta}$ in $\theta \in \mathsf{BT}_{\mathsf{pa(ID)}}$.
2. Define $\mathbf{P}_1(\mathsf{KGC}), \mathbf{P}_2(\mathsf{KGC})$ as the identity matrix $\mathbf{I}_{m\times m}$, and then run
$$\begin{cases} \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\widetilde{\mathsf{ID}})} \leftarrow \mathsf{BasisDel}(\mathbf{A} \cdot \mathbf{P}_1(\mathsf{pa(ID)}),\ \mathbf{H}_1(\widetilde{\mathsf{ID}}),\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{pa(ID)})},\ \sigma_{\ell-1}), \\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})} \leftarrow \mathsf{BasisDel}(\mathbf{A} \cdot \mathbf{P}_1(\mathsf{pa(ID)}),\ \mathbf{H}_1(\mathsf{ID}),\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{pa(ID)})},\ \sigma_{\ell-1}), \\ \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})} \leftarrow \mathsf{BasisDel}(\mathbf{B} \cdot \mathbf{P}_2(\mathsf{pa(ID)}),\ \mathbf{H}_2(\mathsf{ID}),\ \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{pa(ID)})},\ \sigma_{\ell-1}). \end{cases}$$
3. Run $\mathbf{e}_{\mathsf{ID},\theta} \leftarrow \mathsf{SamplePre}(\mathbf{A}\cdot\mathbf{P}_1(\widetilde{\mathsf{ID}}), \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\widetilde{\mathsf{ID}})}, \mathbf{u}_{\mathsf{pa(ID)},\theta}, \tau_{\ell-1})$ for $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$.
4. Create a new binary tree $\mathsf{BT}_{\mathsf{ID}}$ with N leaf nodes.
5. Output $\mathsf{SK}_{\mathsf{ID}} := \Big(\mathsf{BT}_{\mathsf{ID}},\ (\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})},\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})},\ \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})}\Big)$, $\mathsf{SK}'_{\mathsf{pa(ID)}}$.

KeyUp$(\mathsf{PP}, \mathsf{t}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}) \to (\mathsf{KU}_{\mathsf{ID},\mathsf{t}}, \mathsf{SK}'_{\mathsf{ID}})$:

For $|\mathsf{ID}| = \ell \in [0, L-1]$, it performs the following steps.

1. Select $\mathbf{u}_{\mathsf{ID},\theta} \xleftarrow{\$} \mathbb{Z}_q^n$ for node $\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})$, if $\mathbf{u}_{\mathsf{ID},\theta}$ is undefined. Here ID may update $\mathsf{SK}_{\mathsf{ID}}$ to $\mathsf{SK}'_{\mathsf{ID}}$ by storing new defined $\mathbf{u}_{\mathsf{ID},\theta}$ in $\theta \in \mathsf{BT}_{\mathsf{ID}}$.
2. Run $\mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID}\|\mathsf{t})} \leftarrow \mathsf{BasisDel}(\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID}), \mathbf{H}_2(\mathsf{ID}\|\mathsf{t}), \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})}, \sigma_\ell)$, and then run $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta}$ $\leftarrow \mathsf{SamplePre}(\mathbf{B} \cdot \mathbf{P}_2(\mathsf{ID}\|\mathsf{t}), \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID}\|\mathsf{t})}, \mathbf{u} - \mathbf{u}_{\mathsf{ID},\theta}, \tau_\ell)$ for $\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})$.
3. If $\ell \geqslant 1$, run $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} \leftarrow$ **GenDK**$(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}})$, where **GenDK**$(\cdot)$ is defined below. Then extract $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$ from $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$.
4. Output $\mathsf{KU}_{\mathsf{ID},\mathsf{t}} := \Big((\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})},\ (\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}\Big)$, $\mathsf{SK}'_{\mathsf{ID}}$.

GenDK$(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}) \to \mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ or $\perp$:

For $|\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Extract $\mathsf{P} := \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$ in $\mathsf{SK}_{\mathsf{ID}}$, and $\mathsf{K} := \mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa(ID)}}, \mathsf{RL}_{\mathsf{pa(ID)},\mathsf{t}})$ in $\mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}$. If $\mathsf{P} \cap \mathsf{K} = \emptyset$, output $\perp$. Otherwise, for the unique node $\theta^* \in \mathsf{P} \cap \mathsf{K}$, extract $\mathbf{e}_{\mathsf{ID},\theta^*}, \mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} \in \mathbb{Z}^m$ in $\mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}$, respectively. Then set $\mathbf{d}_{\mathsf{ID},\mathsf{t}} := [\mathbf{e}_{\mathsf{ID},\theta^*} \| \mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*}] \in \mathbb{Z}^{2m}$.
2. If $\ell \geqslant 2$, extract $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell-1]}$ from $\mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}$.
3. Run $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID}\|\mathsf{t})} \leftarrow \mathsf{BasisDel}(\mathbf{A} \cdot \mathbf{P}_1(\mathsf{ID}),\ \mathbf{H}_1(\mathsf{ID}\|\mathsf{t}),\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})},\ \sigma_\ell)$, and then run $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \leftarrow \mathsf{SamplePre}(\mathbf{A} \cdot \mathbf{P}_1(\mathsf{ID}\|\mathsf{t}),\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID}\|\mathsf{t})},\ \mathbf{u},\ \tau_\ell)$.
4. Output $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} := \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$.

Decrypt$(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{t}}, \mathsf{CT}) \to \mathsf{M}$:

For $|\mathsf{ID}| = \ell \in [L]$, it performs the following steps.

1. Compute $c' := c_0 - \sum_{i=1}^{\ell} \mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}^{\top}[\mathbf{c}_{i,1} \| \mathbf{c}_{i,2}] - \mathbf{g}_{\mathsf{ID},\mathsf{t}}^{\top}\mathbf{c}_{L+1} \ \in \mathbb{Z}_q$. Treat c' as an integer in $[q] \subset \mathbb{Z}$.
2. Output $\mathsf{M} := 1$ if $|c' - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$, and output $\mathsf{M} := 0$ otherwise.

Correctness. Assume that ID has the depth $|\mathsf{ID}| = \ell \in [L]$. If $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, then one can obtain $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} = \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}} \Big)$. Recall that $\mathbf{d}_{\mathsf{ID},\mathsf{t}} = [\mathbf{e}_{\mathsf{ID},\theta^*} \| \mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*}] \in \mathbb{Z}^{2m}$, where $\theta^* \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}}) \cap \mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa(ID)}}, \mathsf{RL}_{\mathsf{pa(ID)},\mathsf{t}})$. According to

$$[\mathbf{A} \cdot \mathbf{P}_1(\widetilde{\mathsf{ID}})]\mathbf{e}_{\mathsf{ID},\theta^*} = \mathbf{u}_{\mathsf{pa(ID)},\theta^*}, \quad [\mathbf{B} \cdot \mathbf{P}_2(\mathsf{pa(ID)} \| \mathsf{t})]\mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta^*} = \mathbf{u} - \mathbf{u}_{\mathsf{pa(ID)},\theta^*},$$

one can obtain $\mathbf{d}_{\mathsf{ID},\mathsf{t}}^{\top}[\mathbf{c}_{\ell,1} \| \mathbf{c}_{\ell,2}] = \mathbf{u}^{\top}\mathbf{s}_{\ell} + \mathbf{d}_{\mathsf{ID},\mathsf{t}}^{\top}[\mathbf{x}_{\ell,1} \| \mathbf{x}_{\ell,2}]$. Similarly, for $i \in [\ell - 1]$ we also have $\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}^{\top}[\mathbf{c}_{i,1} \| \mathbf{c}_{i,2}] = \mathbf{u}^{\top}\mathbf{s}_i + \mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}^{\top}[\mathbf{x}_{i,1} \| \mathbf{x}_{i,2}]$. Besides, the vector $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \in \mathbb{Z}^m$ satisfies $[\mathbf{A} \cdot \mathbf{P}_1(\mathsf{ID} \| \mathsf{t})]\mathbf{g}_{\mathsf{ID},\mathsf{t}} = \mathbf{u}$, $\mathbf{g}_{\mathsf{ID},\mathsf{t}}^{\top}\mathbf{c}_{L+1} = \mathbf{u}^{\top}\mathbf{s}_{L+1} + \mathbf{g}_{\mathsf{ID},\mathsf{t}}^{\top}\mathbf{x}_{L+1}$. From the above, we can compute

$$\begin{aligned} c' &= \mathbf{u}^{\top}(\textstyle\sum_{i=1}^{\ell} \mathbf{s}_i) + \mathbf{u}^{\top}\mathbf{s}_{L+1} + x + \mathsf{M}\lfloor \frac{q}{2} \rfloor - \sum_{i=1}^{\ell} \mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}^{\top}[\mathbf{c}_{i,1} \| \mathbf{c}_{i,2}] - \mathbf{g}_{\mathsf{ID},\mathsf{t}}^{\top}\mathbf{c}_{L+1} \\ &= \mathsf{M}\lfloor \tfrac{q}{2} \rfloor + (x - \textstyle\sum_{i=1}^{\ell} \mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}^{\top}[\mathbf{x}_{i,1} \| \mathbf{x}_{i,2}] - \mathbf{g}_{\mathsf{ID},\mathsf{t}}^{\top}\mathbf{x}_{L+1}). \end{aligned}$$

Set $z := x - \sum_{i=1}^{\ell} \mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}^{\top}[\mathbf{x}_{i,1} \| \mathbf{x}_{i,2}] - \mathbf{g}_{\mathsf{ID},\mathsf{t}}^{\top}\mathbf{x}_{L+1}$ as the noise. Then according to the triangle inequality, the Cauchy-Schwarz inequality, and Lemma 1, the noise z can be bounded as follows with overwhelming probability:

$$\begin{aligned} \big|z\big| &= \big|x - \textstyle\sum_{i=1}^{\ell}(\mathbf{e}_{\mathsf{ID}_{[i]},\theta^*}^{\top}\mathbf{x}_{i,1} + \mathbf{e}_{\mathsf{pa}(\mathsf{ID}_{[i]}),\mathsf{t},\theta^*}^{\top}\mathbf{x}_{i,2}) - \mathbf{g}_{\mathsf{ID},\mathsf{t}}^{\top}\mathbf{x}_{L+1}\big| \\ &\leqslant \big|x\big| + \textstyle\sum_{i=1}^{\ell}(\big\|\mathbf{e}_{\mathsf{ID}_{[i]},\theta^*}\big\| \cdot \big\|\mathbf{x}_{i,1}\big\| + \big\|\mathbf{e}_{\mathsf{pa}(\mathsf{ID}_{[i]}),\mathsf{t},\theta^*}\big\| \cdot \big\|\mathbf{x}_{i,2}\big\|) + \big\|\mathbf{g}_{\mathsf{ID},\mathsf{t}}\big\| \cdot \big\|\mathbf{x}_{L+1}\big\| \\ &\leqslant \alpha q + \textstyle\sum_{i=1}^{\ell}(\tau_{i-1}\sqrt{m} \cdot \alpha q\sqrt{m} + \tau_{i-1}\sqrt{m} \cdot \alpha q\sqrt{m}) + \tau_{\ell}\sqrt{m} \cdot \alpha q\sqrt{m} \\ &= [1 + (\textstyle\sum_{i=1}^{\ell} 2\sigma_{i-1} + \sigma_{\ell})m^{\frac{3}{2}}\omega(\sqrt{\log m})]\alpha q \\ &\leqslant [1 + (2L+1)\sigma_L m^{\frac{3}{2}}\omega(\sqrt{\log m})]\alpha q \\ &= O(L\sigma_L m^{\frac{3}{2}}\omega(\sqrt{\log m})\alpha q). \end{aligned}$$

Note that in the above $\mathbf{e}_{\mathsf{ID}_{[i]},\theta^*}, \mathbf{e}_{\mathsf{pa}(\mathsf{ID}_{[i]}),\mathsf{t},\theta^*}$, the node θ^* is determined by $\mathsf{ID}_{[i]}, \mathsf{t}$ as shown in the decryption key generation algorithm. As a conclusion, if $O(L\sigma_L m^{\frac{3}{2}}\omega(\sqrt{\log m})\alpha q) < q/5$, we know that $\big|z\big|$ is upper bounded by $q/5$ with overwhelming probability, and thus our RHIBE scheme $\mathbf{\Pi}_2$ only has negligible decryption error.

Parameters. The analysis for parameters selection is similar to that in Section 3.1. We must consider the condition $O(L\sigma_L m^{\frac{3}{2}}\omega(\sqrt{\log m})\alpha q) < q/5$ for the correctness requirement, and the condition $q > 2\sqrt{n}/\alpha$ for the hardness assumption of $\mathsf{LWE}_{n,2m+1,q,\mathcal{D}_{\mathbb{Z}^{2m+1},\alpha q}}$. Besides, we also need to make sure that algorithms such as $\mathsf{BasisDel}$ et al. can operate in the construction, and algorithms such as $\mathsf{SampleRwithBasis}$ et al. can work in the security proof. Finally, we set the parameters used for our RHIBE scheme $\mathbf{\Pi}_2$ as follows:

$$\begin{aligned} &m = 6n^{1+\delta} = O(Ln\log n), && \alpha = [Lm^{\frac{3}{2}L+3}\omega(\log^{2L+\frac{5}{2}} n)]^{-1}, \\ &q = Lm^{\frac{3}{2}L+3}n^{\frac{1}{2}}\omega(\log^{2L+\frac{5}{2}} n), && \sigma_{\ell} = m^{\frac{3}{2}\ell+\frac{3}{2}}\omega(\log^{2\ell+2} n) \ \text{ for } \ \ell \in [0, L], \end{aligned}$$

and round up m to the nearest larger integer, and q to the nearest larger prime. Here we choose δ such that $n^{\delta} > \lceil \log q \rceil = O(L \log n)$.

4.2 Security

Theorem 2. *The RHIBE scheme $\mathbf{\Pi}_2$ satisfies the adaptive-identity security, assuming the hardness of the problem $\mathsf{LWE}_{n,2m+1,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^{2m+1},\alpha q}$.*

It is shown in Section 3.2 that the attack strategies taken by $\mathcal{A}$ can be divided into the **Type-I** strategy (further divided into **Type-I-i^***) and the **Type-II** strategy. In the security game, we separately describe the progress for these two types of attack strategies. Then according to the "strategy-dividing lemma" introduced in [11,12], we can complete the proof of Theorem 2. First of all, for an adversary $\mathcal{A}$ that uses the **Type-I** strategy (instead of the **Type-I-i^*** strategy in Lemma 3), we have the following result.

Lemma 5. *Suppose that a PPT adversary $\mathcal{A}$ follows the* **Type-I** *strategy, and its adaptive-identity security advantage is denoted by $\mathsf{Adv}^{\textbf{Type-I}}_{\mathbf{\Pi}_2,L,\mathcal{A}}(n)$. Besides, let $Q_{\mathbf{H}_1}, Q_{\mathbf{H}_2}$ denote the maximum numbers of queries made by $\mathcal{A}$ to the random oracles $\mathbf{H}_1, \mathbf{H}_2$, respectively. Then there exits a PPT algorithm $\mathcal{C}$, whose advantage for the $\mathsf{LWE}_{n,2m+1,q,\chi}$ ($\chi = \mathcal{D}_{\mathbb{Z}^{2m+1},\alpha q}$) problem is denoted by $\mathsf{Adv}^{\mathsf{LWE}}_{\mathcal{C}}(n)$, such that*

$$\mathsf{Adv}^{\textbf{Type-I}}_{\mathbf{\Pi}_2,L,\mathcal{A}}(n) \leqslant (2L \cdot Q^L_{\mathbf{H}_1} \cdot Q_{\mathbf{H}_2}) \cdot \mathsf{Adv}^{\mathsf{LWE}}_{\mathcal{C}}(n) + \mathsf{negl}(n).$$

Proof. The algorithm $\mathcal{C}$, which we are going to construct, simulates an attack environment for the adversary $\mathcal{A}$ that uses the **Type-I** strategy as follows.

Instance. $\mathcal{C}$ is given the problem instance of $\mathsf{LWE}_{n,2m+1,q,\chi}$ as $(\widehat{\mathbf{A}}, \widehat{\mathbf{v}}) \in \mathbb{Z}_q^{n\times(2m+1)} \times \mathbb{Z}_q^{2m+1}$ for $\widehat{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{n\times(2m+1)}$. The task of $\mathcal{C}$ is to distinguish whether (1) $\widehat{\mathbf{v}} = \widehat{\mathbf{A}}^\top \widehat{\mathbf{s}} + \widehat{\mathbf{x}}$ for some $\widehat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_q^n$ and some $\widehat{\mathbf{x}} \hookleftarrow \chi$, or (2) $\widehat{\mathbf{v}} \xleftarrow{\$} \mathbb{Z}_q^{2m+1}$. Here we assume that $\widehat{\mathbf{A}} = [\ \mathbf{a}_0 \mid \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_{2m}\ ] \in \mathbb{Z}_q^{n\times(2m+1)}$ and $\widehat{\mathbf{v}} = (v_0, v_1, \cdots, v_{2m}) \in \mathbb{Z}_q^{2m+1}$.

Setup. Firstly, $\mathcal{C}$ directly selects $i^* \xleftarrow{\$} [L]$. Namely, $\mathcal{C}$ guesses that the strategy taken by the adversary $\mathcal{A}$ is **Type-I-i^***. Then $\mathcal{C}$ chooses $Q_2^* \xleftarrow{\$} [Q_{\mathbf{H}_2}]$, and selects $Q_{1,j}^* \xleftarrow{\$} [Q_{\mathbf{H}_1}]$, $\mathbf{R}_{1,j}^* \hookleftarrow \mathcal{D}_{m\times m}$, $\mathbf{R}_{2,j}^* \hookleftarrow \mathcal{D}_{m\times m}$ for $j \in [i^*]$. Besides, $\mathcal{C}$ sets $\mathbf{u} := \mathbf{a}_0 \in \mathbb{Z}_q^n$, $\mathbf{A}_0 := [\ \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_m\ ] \in \mathbb{Z}_q^{n\times m}$, $\mathbf{B}_0 := [\ \mathbf{a}_{m+1} \mid \cdots \mid \mathbf{a}_{2m}\ ] \in \mathbb{Z}_q^{n\times m}$. Namely, we have $\widehat{\mathbf{A}} = [\ \mathbf{u} \mid \mathbf{A}_0 \mid \mathbf{B}_0\ ]$. Next, $\mathcal{C}$ sets $\mathbf{A} := \mathbf{A}_0(\mathbf{R}_{1,i^*}^* \cdots \mathbf{R}_{1,2}^* \mathbf{R}_{1,1}^*)$, $\mathbf{B} := \mathbf{B}_0(\mathbf{R}_{2,i^*}^* \cdots \mathbf{R}_{2,2}^* \mathbf{R}_{2,1}^*)$. One can check that $\mathbf{A}, \mathbf{B}$ are both uniform in $\mathbb{Z}_q^{n\times m}$, and $\mathbf{u}$ is uniform in $\mathbb{Z}_q^n$. Finally, $\mathcal{C}$ publishes the public parameters $\mathsf{PP} := (\mathbf{A}, \mathbf{B}, \mathbf{u})$. (Note that here $\mathcal{C}$ must also give the key update $\mathsf{KU}_{\mathsf{KGC},1}$ to $\mathcal{A}$. For convenience, the construction of $\mathsf{KU}_{\mathsf{KGC},1}$ will be given later, together with the construction of other key updates.)

Random Oracle Query. For each random oracle, we assume that the queries are unique, otherwise $\mathcal{C}$ simply returns the same output on the same input without incrementing the query counter. Besides, without loss of generality, we can assume that for any $\mathsf{CH} \in (\{0,1,2\}^\omega)^{\leqslant L}$, the $\mathbf{H}_2$ query on CH is preceded by the $\mathbf{H}_1$ query on CH. Then $\mathcal{C}$ answers $\mathcal{A}$'s queries as follows.

Query $\mathbf{H}_1$ on $\mathsf{CH} \in (\{0,1,2\}^\omega)^{\leqslant L+1}$: Suppose that it is the Q_1-th query. (1) If $Q_1 = Q_{1,j}^*$ for some $j \in [i^*]$, define $\mathbf{H}_1(\mathsf{CH}) := \mathbf{R}_{1,j}^*$. (2) Otherwise, let $\ell := |\mathsf{CH}|$ be the depth of CH. If $\ell \leqslant i^* + 1$, compute $\mathbf{F}_{1,\ell} := \mathbf{A} \cdot (\mathbf{R}_{1,\ell-1}^* \cdots \mathbf{R}_{1,2}^* \mathbf{R}_{1,1}^*)^{-1}$, run $(\mathbf{R}, \mathbf{T}_\mathbf{F}) \leftarrow \mathsf{SampleRwithBasis}(\mathbf{F}_{1,\ell}, \sigma_{\ell-1})$ for $\mathbf{F} := \mathbf{F}_{1,\ell}\mathbf{R}^{-1}$, and define $\mathbf{H}_1(\mathsf{CH}) := \mathbf{R}$. Besides, $\mathcal{C}$ saves the tuple $(\ell, \mathsf{CH}, \mathbf{R}, \mathbf{F}, \mathbf{T}_\mathbf{F})$ for future use. (3) If $\ell > i^* + 1$, just select $\mathbf{R} \hookleftarrow \mathcal{D}_{m\times m}$ and then set $\mathbf{H}_1(\mathsf{CH}) := \mathbf{R}$. (4) Finally, $\mathcal{C}$ returns $\mathbf{H}_1(\mathsf{CH})$ to $\mathcal{A}$.

Query $\mathbf{H}_2$ on $\mathsf{CH} \in (\{0,1,2\}^\omega)^{\leqslant L}$: Suppose that it is the Q_2-th query. (1) Let $\mathcal{C}$ check the value of $\mathbf{H}_1(\mathsf{CH})$. If $\mathbf{H}_1(\mathsf{CH}) = \mathbf{R}_{1,j}^*$ holds for some $j \in [i^* - 1]$, then define $\mathbf{H}_2(\mathsf{CH}) := \mathbf{R}_{2,j}^*$. (2) Otherwise, $\mathcal{C}$ checks whether $Q_2 = Q_2^*$ holds. In case $Q_2 = Q_2^*$, define $\mathbf{H}_2(\mathsf{CH}) := \mathbf{R}_{2,i^*}^*$. (3) In case $Q_2 \neq Q_2^*$, let $\ell := |\mathsf{CH}|$ be the depth of CH. If $\ell \leqslant i^* + 1$, compute $\mathbf{F}_{2,\ell} := \mathbf{B} \cdot (\mathbf{R}_{2,\ell-1}^* \cdots \mathbf{R}_{2,2}^* \mathbf{R}_{2,1}^*)^{-1}$, run $(\mathbf{R}, \mathbf{T}_\mathbf{F}) \leftarrow \mathsf{SampleRwithBasis}(\mathbf{F}_{2,\ell}, \sigma_{\ell-1})$ for $\mathbf{F} := \mathbf{F}_{2,\ell}\mathbf{R}^{-1}$, and define $\mathbf{H}_2(\mathsf{CH}) := \mathbf{R}$. Besides, $\mathcal{C}$ saves the tuple $(\ell, \mathsf{CH}, \mathbf{R}, \mathbf{F}, \mathbf{T}_\mathbf{F})$ for future use. (4) If $\ell > i^* + 1$, just select $\mathbf{R} \hookleftarrow \mathcal{D}_{m\times m}$ and then set $\mathbf{H}_2(\mathsf{CH}) := \mathbf{R}$. (5) Finally, $\mathcal{C}$ returns $\mathbf{H}_2(\mathsf{CH})$ to $\mathcal{A}$.

According to the above setup algorithm, $\mathcal{C}$ does not own the trapdoors $\mathbf{T}_\mathbf{A}, \mathbf{T}_\mathbf{B}$. However, due to the above random oracle query, we still have the following result, where the sets

$\mathcal{CH}_1, \mathcal{CH}_2$ are defined as

$$\mathcal{CH}_1 := \left\{ \mathsf{CH} \in (\{0,1,2\}^{\omega})^{\leqslant L+1} \mid |\mathsf{CH}| \leqslant i^*, \text{ and } \mathbf{H}_1(\mathsf{CH}_{[j]}) = \mathbf{R}^*_{1,j} \text{ for } j = 1, 2, \cdots, |\mathsf{CH}| \right\},$$
$$\mathcal{CH}_2 := \left\{ \mathsf{CH} \in (\{0,1,2\}^{\omega})^{\leqslant L} \mid |\mathsf{CH}| \leqslant i^*, \text{ and } \mathbf{H}_2(\mathsf{CH}_{[j]}) = \mathbf{R}^*_{2,j} \text{ for } j = 1, 2, \cdots, |\mathsf{CH}| \right\}.$$

Lemma 6. *The setup algorithm and the random oracle query are shown as above.*

(1) Suppose that for some $\mathsf{CH} \in (\{0,1,2\}^{\omega})^{\leqslant L+1}$ *with* $|\mathsf{CH}| = \ell$*, the adversary* $\mathcal{A}$ *has queried* $\mathbf{H}_1$ *on all* $\mathsf{CH}' \in \mathsf{prefix}(\mathsf{CH})$*. Then if* $\mathsf{CH} \notin \mathcal{CH}_1$*, the algorithm* $\mathcal{C}$ *is able to construct a short basis* $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH})}$ *distributed statistically close to* $\mathcal{D}_{Basis}(\Lambda_q^{\perp}(\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH})), \sigma_{\ell-1})$*.*

(2) Suppose that for some $\mathsf{CH} \in (\{0,1,2\}^{\omega})^{\leqslant L}$ *with* $|\mathsf{CH}| = \ell$*, the adversary* $\mathcal{A}$ *has queried* $\mathbf{H}_2$ *on all* $\mathsf{CH}' \in \mathsf{prefix}(\mathsf{CH})$*. Then if* $\mathsf{CH} \notin \mathcal{CH}_2$*, the algorithm* $\mathcal{C}$ *is able to construct a short basis* $\mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{CH})}$ *distributed statistically close to* $\mathcal{D}_{Basis}(\Lambda_q^{\perp}(\mathbf{B}\cdot\mathbf{P}_2(\mathsf{CH})), \sigma_{\ell-1})$*.*

Proof. (1) Let us define the integer k at first. If $\mathbf{H}_1(\mathsf{CH}_{[j]}) = \mathbf{R}^*_{1,j}$ for all $j \in [\min\{i^*, |\mathsf{CH}|\}]$, then $|\mathsf{CH}| \geqslant i^* + 1$ must hold since $\mathsf{CH} \notin \mathcal{CH}_1$, and we define $k = i^* + 1$ for this case. Otherwise, we define $k \in [\min\{i^*, |\mathsf{CH}|\}]$ as the smallest index such that $\mathbf{H}_1(\mathsf{CH}_{[k]}) \neq \mathbf{R}^*_{1,k}$. Note that $k \leqslant i^* + 1$, and $\mathbf{H}_1(\mathsf{CH}_{[j]}) = \mathbf{R}^*_{1,j}$ for all $j \in [k-1]$. From the $\mathbf{H}_1$ query history, $\mathcal{C}$ retrieves the saved tuple $(k, \mathsf{CH}_{[k]}, \mathbf{R}, \mathbf{F}, \mathbf{T}_{\mathbf{F}})$, which is created when the adversary $\mathcal{A}$ queries $\mathbf{H}_1$ on $\mathsf{CH}_{[k]}$. By construction, we know $\mathbf{F} = \mathbf{A} \cdot (\mathbf{R}^*_{1,1})^{-1}(\mathbf{R}^*_{1,2})^{-1} \cdots (\mathbf{R}^*_{1,k-1})^{-1}\mathbf{R}^{-1} = \mathbf{A} \cdot (\mathbf{H}_1(\mathsf{CH}_{[1]}))^{-1}(\mathbf{H}_1(\mathsf{CH}_{[2]}))^{-1} \cdots (\mathbf{H}_1(\mathsf{CH}_{[k-1]}))^{-1}(\mathbf{H}_1(\mathsf{CH}_{[k]}))^{-1} = \mathbf{A} \cdot \mathbf{P}_1(\mathsf{CH}_{[k]})$, and $\mathbf{T}_{\mathbf{F}} = \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH}_{[k]})}$ is distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^{\perp}(\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH}_{[k]})), \sigma_{k-1})$. If $k = \ell$, the basis $\mathbf{T}_{\mathbf{F}}$ is already the desired $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH})}$. If $k < \ell$, $\mathcal{C}$ runs $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH}_{[j]})} \leftarrow \mathsf{BasisDel}(\mathbf{A} \cdot \mathbf{P}_1(\mathsf{CH}_{[j-1]}), \mathbf{H}_1(\mathsf{CH}_{[j]}), \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH}_{[j-1]})}, \sigma_{j-1})$ for $j = k{+}1, k{+}2, \cdots, \ell$, and finally the desired $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH})}$ is also obtained.

(2) The proof of this part is the same as that of (1). □

Assume that ID^*, t^* are the challenge identity and time period, which $\mathcal{A}$ sends to $\mathcal{C}$ in the challenge query. Then we let $\mathsf{Success}$ be the event that the adversary $\mathcal{A}$ follows the **Type-I-**i^* strategy (which implicitly implies $|\mathsf{ID}^*| \geqslant i^*$), and $\widetilde{\mathsf{ID}^*_{[i^*]}} \in \mathcal{CH}_1$, $\mathsf{ID}^*_{[i^*-1]} \| \mathsf{t}^* \in \mathcal{CH}_2$ holds, and $\mathcal{C}$ does not fail due to collisions on $\mathbf{H}_1$ or $\mathbf{H}_2$ found by $\mathcal{A}$. According to $\mathsf{pa}(\widetilde{\mathsf{ID}^*_{[i^*]}}) = \mathsf{pa}(\mathsf{ID}^*_{[i^*-1]} \| \mathsf{t}^*) = \mathsf{ID}^*_{[i^*-1]}$ and the above $\mathbf{H}_2$ random oracle query, we have $\widetilde{\mathsf{ID}^*_{[i^*]}} \in \mathcal{CH}_1$, $\mathsf{ID}^*_{[i^*-1]} \| \mathsf{t}^* \in \mathcal{CH}_2 \;\Leftrightarrow\; \widetilde{\mathsf{ID}^*_{[i^*]}} \in \mathcal{CH}_1$, $\mathbf{H}_2(\mathsf{ID}^*_{[i^*-1]} \| \mathsf{t}^*) = \mathbf{R}^*_{2,i^*}$. Therefore, we obtain $\Pr[\mathsf{Success}] = 1/L \cdot 1/Q_{\mathbf{H}_1}^{i^*} \cdot 1/Q_{\mathbf{H}_2} \cdot (1 - \mathsf{negl}(n))$.

In the following, we show that if $\mathsf{Success}$ happens, the algorithm $\mathcal{C}$ will successfully simulate the attack environment for $\mathcal{A}$. Otherwise, $\mathcal{C}$ will fail and abort at some point.

Suppose that the event $\mathsf{Success}$ happens. In order to respond to the secret key generation query, the secret key reveal query, the revoke & key update query, and the decryption key reveal query, which are made by $\mathcal{A}$ using the **Type-I-**i^* strategy, $\mathcal{C}$ must construct the following items:

- (a) $\mathsf{SK}_{\mathsf{ID}} = \Big(\mathsf{BT}_{\mathsf{ID}},\ (\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID})}, \eta_{\mathsf{ID}})},\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})},\ \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})}\Big)$ for $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L} \setminus \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]})$;
- (b) $\mathsf{KU}_{\mathsf{ID},\mathsf{t}} = \Big((\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}}, \mathsf{RL}_{\mathsf{ID},\mathsf{t}})},\ (\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell]}\Big)$ for $\mathsf{ID} \in \{\mathsf{KGC}\} \cup (\mathcal{ID})^{\leqslant L-1}$, $\mathsf{t} \in \mathcal{T}$ and $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, where $\ell = |\mathsf{ID}|$;
- (c) $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} = \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i \in [\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$ for $(\mathsf{ID},\mathsf{t}) \in (\mathcal{ID})^{\leqslant L} \times \mathcal{T} \setminus \{(\mathsf{ID}^*, \mathsf{t}^*)\}$ and $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, where $\ell = |\mathsf{ID}|$.

$\mathcal{C}$ needs to generate any item in (a) in the secret key generation query and return it to the adversary $\mathcal{A}$ in the secret key reveal query. In the revoke & key update query (and in the secret key generation query, and at the setup), $\mathcal{C}$ must return the corresponding items in (b) to $\mathcal{A}$. Similarly, $\mathcal{A}$ is allowed to query any item in (c) as a decryption key reveal query. Note that these four queries can be made before the challenge query, and in this case $\mathcal{C}$ does not know ID^*, t^* (the challenge identity and time period). However, if $\mathsf{Success}$ happens, we can regard $\mathcal{CH}_1$ as its subset $\mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$, and regard $\mathcal{CH}_2$ as its subset $\mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]}\|\mathsf{t}^*)$. We do not need to consider the case that $\mathcal{A}$ finds some $\mathsf{CH}_1 \in \mathcal{CH}_1 \setminus \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$, or some $\mathsf{CH}_2 \in \mathcal{CH}_2 \setminus \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]}\|\mathsf{t}^*)$, which implies that $\mathcal{A}$ finds collisions on $\mathbf{H}_1$ or $\mathbf{H}_2$. The failure of $\mathcal{C}$ due to this case is directly denied by the definition of the event $\mathsf{Success}$.

As a preparation for the construction of the above (a), (b) and (c), $\mathcal{C}$ must deal with $\mathsf{BT}_{\mathsf{ID}}$ differently for some user ID satisfying $\mathsf{ID} = \mathsf{KGC}$ (if $i^* = 1$), or $|\mathsf{ID}| = i^* - 1, \mathsf{ID} \in \mathcal{CH}_1$ (if $i^* \geqslant 2$). Specifically, $\mathcal{C}$ must change the way that the vectors $(\mathbf{u}_{\mathsf{ID},\theta})_{\theta\in\mathsf{BT}_{\mathsf{ID}}}$ stored in nodes of $\mathsf{BT}_{\mathsf{ID}}$ are generated. Once $\mathsf{BT}_{\mathsf{ID}}$ is created at the setup or in the secret key generation query, $\mathcal{C}$ randomly pick a leaf node η^* from $\mathsf{BT}_{\mathsf{ID}}$. In the future, η^* is only used to store some user ID_{R} which has ID as its parent and satisfies $\mathbf{H}_1(\widetilde{\mathsf{ID}_{\mathsf{R}}}) = \mathbf{R}^*_{1,i^*}$. This can be done, since $\mathcal{C}$ can check whether $\mathbf{H}_1(\widetilde{\mathsf{ID}_{\mathsf{ch}}}) = \mathbf{R}^*_{1,i^*}$ holds before assigning a leaf node of $\mathsf{BT}_{\mathsf{ID}}$ to ID's child user $\mathsf{ID}_{\mathsf{ch}}$. Note that if the event $\mathsf{Success}$ happens, the above $\mathsf{ID}, \mathsf{ID}_{\mathsf{R}}$ will denote $\mathsf{pa}(\mathsf{ID}^*_{[i^*]}), \mathsf{ID}^*_{[i^*]}$, respectively. When $\mathbf{u}_{\mathsf{ID},\theta}$ for some $\theta \in \mathsf{BT}_{\mathsf{ID}}$ must be defined for $\mathcal{C}$ to answer $\mathcal{A}$'s query, $\mathcal{C}$ proceeds as follows. If $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{ID}}, \eta^*)$, $\mathcal{C}$ first samples $\mathbf{e}_{1,\theta} \hookleftarrow \mathcal{D}_{\mathbb{Z}^m,\tau_{i^*-1}}$, and then sets $\mathbf{u}_{\mathsf{ID},\theta} := \mathbf{A}_0 \cdot \mathbf{e}_{1,\theta}$. If $\theta \in \mathsf{BT}_{\mathsf{ID}} \setminus \mathsf{Path}(\mathsf{BT}_{\mathsf{ID}}, \eta^*)$, $\mathcal{C}$ first samples $\mathbf{e}_{2,\theta} \hookleftarrow \mathcal{D}_{\mathbb{Z}^m,\tau_{i^*-1}}$, and then sets $\mathbf{u}_{\mathsf{ID},\theta} := \mathbf{u} - \mathbf{B}_0 \cdot \mathbf{e}_{2,\theta}$. $\mathcal{C}$ keeps the obtained $\mathbf{e}_{1,\theta}$ or $\mathbf{e}_{2,\theta}$ secret for future use.

Construction of $\mathsf{SK}_{\mathsf{ID}}$. Undoubtedly, $\mathcal{C}$ is able to construct the item $\mathsf{BT}_{\mathsf{ID}}$ for any $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L}$. Now let us see the generation of the items $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})}$ and $\mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})}$. According to Lemma 6, if $\mathsf{ID} \notin \mathcal{CH}_1$, $\mathcal{C}$ is able to construct the short basis $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})}$. Similarly, if $\mathsf{ID} \notin \mathcal{CH}_2$, the item $\mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})}$ can also be obtained. Besides, we let $\mathcal{C}$ use the symbol $\perp$ to denote $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID})}$ for $\mathsf{ID} \in \mathcal{CH}_1$ and $\mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID})}$ for $\mathsf{ID} \in \mathcal{CH}_2$. The symbol $\perp$ denotes that the corresponding item in $\mathsf{SK}_{\mathsf{ID}}$ cannot be constructed. If there is a symbol $\perp$ in $\mathsf{SK}_{\mathsf{ID}}$ created in the secret key generation query, $\mathcal{C}$ fails and aborts only when $\mathcal{C}$ must return the same $\mathsf{SK}_{\mathsf{ID}}$ to $\mathcal{A}$ in the secret key reveal query.

As for the item $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$, $\mathcal{C}$ needs to consider $\widetilde{\mathsf{ID}}$. (1) If $\widetilde{\mathsf{ID}} \notin \mathcal{CH}_1$, $\mathcal{C}$ is able to construct the short basis $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\widetilde{\mathsf{ID}})}$ due to Lemma 6. Then $\mathcal{C}$ can obtain $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$ by running $\mathbf{e}_{\mathsf{ID},\theta} \leftarrow \mathsf{SamplePre}(\mathbf{A}\cdot\mathbf{P}_1(\widetilde{\mathsf{ID}}),\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\widetilde{\mathsf{ID}})},\ \mathbf{u}_{\mathsf{pa(ID)},\theta},\ \tau_{\ell-1})$ for $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$. (2) If $\widetilde{\mathsf{ID}} \in \mathcal{CH}_1$ and $|\mathsf{ID}| = i^*$, we know that $\mathsf{pa(ID)} = \mathsf{KGC}$ (if $i^* = 1$), or $|\mathsf{pa(ID)}| = i^* - 1, \mathsf{pa(ID)} \in \mathcal{CH}_1$ (if $i^* \geqslant 2$). Thus $\mathcal{C}$ must deal with $\mathsf{BT}_{\mathsf{pa(ID)}}$ differently, which is shown as before. According to Lemma 2 and the fact $\mathbf{u}_{\mathsf{ID},\theta} = \mathbf{A}_0 \cdot \mathbf{e}_{1,\theta} = \mathbf{A}_0(\mathbf{R}^*_{1,i^*}\cdots\mathbf{R}^*_{1,2}\mathbf{R}^*_{1,1}) \cdot (\mathbf{R}^*_{1,i^*}\cdots\mathbf{R}^*_{1,2}\mathbf{R}^*_{1,1})^{-1} \cdot \mathbf{e}_{1,\theta} = \mathbf{A} \cdot \mathbf{P}_1(\widetilde{\mathsf{ID}}) \cdot \mathbf{e}_{1,\theta}$ where $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{ID}}, \eta^*)$, we know that $\mathcal{C}$ is able to obtain $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$ after setting $\eta^* := \eta_{\mathsf{ID}}$ and $\mathbf{e}_{\mathsf{ID},\theta} := \mathbf{e}_{1,\theta}$. (3) If $\widetilde{\mathsf{ID}} \in \mathcal{CH}_1$ and $|\mathsf{ID}| < i^*$, $\mathcal{C}$ just uses the symbol $\perp$ to denote $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$.

The above construction of $\mathsf{SK}_{\mathsf{ID}}$ is made in the secret key generation query. When $\mathcal{C}$ must return $\mathsf{SK}_{\mathsf{ID}}$ to $\mathcal{A}$ in the secret key reveal query, $\mathcal{C}$ fails and aborts if there is a symbol $\perp$ in $\mathsf{SK}_{\mathsf{ID}}$. However, assuming the occurrence of $\mathsf{Success}$, $\mathcal{C}$ will never fail. The event $\mathsf{Success}$ lets anyone regard $\mathcal{CH}_1$ as $\mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$, and $\mathcal{CH}_2$ as $\mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]}\|\mathsf{t}^*)$. From the above (a) we know $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L} \setminus \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]})$, which implies $\mathsf{ID} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$ and $\mathsf{ID} \notin \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]}\|\mathsf{t}^*)$ due to $\mathcal{ID} \cap \widetilde{\mathcal{ID}} = \emptyset$ and $\mathcal{ID} \cap \mathcal{T} = \emptyset$, respectively. Thus we can regard that $\mathsf{ID} \notin \mathcal{CH}_1$ and $\mathsf{ID} \notin \mathcal{CH}_2$ always holds. Similarly, due to $\widetilde{\mathcal{ID}} \cap \mathcal{ID} = \emptyset$, we have $\widetilde{\mathsf{ID}} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$ for $|\mathsf{ID}| < i^*$, and thus $\widetilde{\mathsf{ID}} \notin \mathcal{CH}_1$ always holds for $|\mathsf{ID}| < i^*$. Note that here the successful construction of $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$ does not need the condition $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L} \setminus \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]})$. Thus if $\mathsf{Success}$ happens, we claim that the item $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$ in $\mathsf{SK}_{\mathsf{ID}}$ can be constructed by $\mathcal{C}$ for any $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L}$, which will be important for the construction of $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$ in $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ and $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$.

Construction of $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$. In order to create the item $(\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})}$, $\mathcal{C}$ should consider $\mathsf{ID}\|\mathsf{t}$. (1) If $\mathsf{ID}\|\mathsf{t} \notin \mathcal{CH}_2$, $\mathcal{C}$ is able to construct the short basis $\mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID}\|\mathsf{t})}$ due to Lemma 6. Then $\mathcal{C}$ can obtain $(\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})}$ by running $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} \leftarrow \mathsf{SamplePre}(\mathbf{B}\cdot$

$\mathbf{P}_2(\mathsf{ID}\|\mathsf{t}),\ \mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID}\|\mathsf{t})},\ \mathbf{u}-\mathbf{u}_{\mathsf{ID},\theta},\ \tau_\ell)$ for $\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})$. (2) If $\mathsf{ID}\|\mathsf{t} \in \mathcal{CH}_2$ and $|\mathsf{ID}\|\mathsf{t}| = i^*$, we know that $\mathsf{ID} = \mathsf{KGC}$ (if $i^* = 1$), or $|\mathsf{ID}| = i^* - 1, \mathsf{ID} \in \mathcal{CH}_2, \mathsf{ID} \in \mathcal{CH}_1$ (if $i^* \geqslant 2$) due to the $\mathbf{H}_2$ random oracle query. Thus $\mathcal{C}$ must deal with $\mathsf{BT}_{\mathsf{ID}}$ differently, which is shown as before. Then $\mathcal{C}$ checks whether $\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}}) \subseteq \mathsf{BT}_{\mathsf{ID}}\backslash\mathsf{Path}(\mathsf{BT}_{\mathsf{ID}},\eta^*)$ holds. If not, $\mathcal{C}$ uses the symbol $\perp$ to denote $(\theta,\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})}$. Otherwise, according to Lemma 2 and the fact $\mathbf{u}-\mathbf{u}_{\mathsf{ID},\theta} = \mathbf{B}_0\cdot\mathbf{e}_{2,\theta} = \mathbf{B}_0(\mathbf{R}^*_{2,i^*}\cdots\mathbf{R}^*_{2,2}\mathbf{R}^*_{2,1})\cdot(\mathbf{R}^*_{2,i^*}\cdots\mathbf{R}^*_{2,2}\mathbf{R}^*_{2,1})^{-1}\cdot\mathbf{e}_{2,\theta} = \mathbf{B}\cdot\mathbf{P}_2(\mathsf{ID}\|\mathsf{t})\cdot\mathbf{e}_{2,\theta}$ where $\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})$. we know that $\mathcal{C}$ is able to obtain $(\theta,\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})}$ after setting $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta} := \mathbf{e}_{2,\theta}$. (3) If $\mathsf{ID}\|\mathsf{t} \in \mathcal{CH}_2$ and $|\mathsf{ID}\|\mathsf{t}| < i^*$, $\mathcal{C}$ just uses the symbol $\perp$ to denote $(\theta,\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})}$.

Assuming that $(\theta,\mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$ in $\mathsf{SK}_{\mathsf{ID}}$ and $(\theta,\mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa(ID)}},\mathsf{RL}_{\mathsf{pa(ID)},\mathsf{t}})}$ in $\mathsf{KU}_{\mathsf{pa(ID)},\mathsf{t}}$ are successfully constructed using the above way, $\mathcal{C}$ can use these two items to generate $\mathbf{d}_{\mathsf{ID},\mathsf{t}}$ for $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, just as the algorithm **GenDK** does. A symbol $\perp$ for either of these two items, directly implies a symbol $\perp$ for $\mathbf{d}_{\mathsf{ID},\mathsf{t}}$. Furthermore, $\mathcal{C}$ uses the symbol $\perp$ to denote $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$, when there is some $\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}}$ denoted by $\perp$.

When $\mathcal{C}$ must return $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ to $\mathcal{A}$, $\mathcal{C}$ fails and aborts if there is a symbol $\perp$ in $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$. However, if $\mathsf{Success}$ happens, in the following we show that $\mathcal{C}$ will never fail. First of all, we can regard the case of $\mathsf{ID}\|\mathsf{t} \in \mathcal{CH}_2, |\mathsf{ID}\|\mathsf{t}| = i^*$ as the case of $\mathsf{ID} = \mathsf{ID}^*_{[i^*-1]}, \mathsf{t} = \mathsf{t}^*$. Besides, the adversary $\mathcal{A}$ must issue a secret key reveal query on $\mathsf{ID}^*_{[i^*]}$, and thus we have $\mathsf{ID}^*_{[i^*]} \in \mathsf{RL}_{\mathsf{t}^*}$ according to the security definition. When $\mathsf{ID}^*_{[i^*-1]} \in \mathsf{RL}_{\mathsf{t}^*}$, $\mathcal{C}$ does not need to construct $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ for $\mathsf{ID} = \mathsf{ID}^*_{[i^*-1]}, \mathsf{t} = \mathsf{t}^*$. When $\mathsf{ID}^*_{[i^*-1]} \notin \mathsf{RL}_{\mathsf{t}^*}, \mathsf{ID}^*_{[i^*]} \in \mathsf{RL}_{\mathsf{ID}^*_{[i^*-1]},\mathsf{t}^*}$, the condition $\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}}) \subseteq \mathsf{BT}_{\mathsf{ID}} \setminus \mathsf{Path}(\mathsf{BT}_{\mathsf{ID}},\eta^*)$ must hold for $\mathsf{ID} = \mathsf{ID}^*_{[i^*-1]}, \mathsf{t} = \mathsf{t}^*, \eta^* = \eta_{\mathsf{ID}^*_{[i^*]}}$. Moreover, due to $\mathcal{T} \cap \mathcal{ID} = \emptyset$, we have $\mathsf{ID}\|\mathsf{t} \notin \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]}\|\mathsf{t}^*)$ for $|\mathsf{ID}\|\mathsf{t}| < i^*$, and thus $\mathsf{ID}\|\mathsf{t} \notin \mathcal{CH}_2$ always holds for $|\mathsf{ID}\|\mathsf{t}| < i^*$. Finally, when $\mathsf{Success}$ happens, the successful construction of $(\theta,\mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa(ID)}},\eta_{\mathsf{ID}})}$ and $(\theta,\mathbf{e}_{\mathsf{pa(ID)},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa(ID)}},\mathsf{RL}_{\mathsf{pa(ID)},\mathsf{t}})}$ directly implies the successful construction of $\mathbf{d}_{\mathsf{ID},\mathsf{t}}$ for $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$.

Construction of $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$. The generation of $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$ in $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ is similar to that in $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$. As for the item $\mathbf{g}_{\mathsf{ID},\mathsf{t}}$, if $\mathsf{ID}\|\mathsf{t} \notin \mathcal{CH}_1$, $\mathcal{C}$ is able to construct the short basis $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID}\|\mathsf{t})}$ due to Lemma 6. Then $\mathcal{C}$ runs $\mathbf{g}_{\mathsf{ID},\mathsf{t}} \leftarrow \mathsf{SamplePre}(\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID}\|\mathsf{t}),\ \mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{ID}\|\mathsf{t})},\ \mathbf{u},\ \tau_\ell)$. If $\mathsf{ID}\|\mathsf{t} \in \mathcal{CH}_1$, $\mathcal{C}$ just uses the symbol $\perp$ to denote $\mathbf{g}_{\mathsf{ID},\mathsf{t}}$. Similar to before, when $\mathcal{C}$ must return $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ to $\mathcal{A}$, $\mathcal{C}$ fails and aborts if there is a symbol $\perp$ in $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$. Suppose that $\mathsf{Success}$ happens. Then we always have $\mathsf{ID}\|\mathsf{t} \notin \mathcal{CH}_1$, since $\mathsf{ID}\|\mathsf{t} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$ always holds due to $\mathcal{T}\cap\mathcal{ID} = \mathcal{T}\cap\widetilde{\mathcal{ID}} = \emptyset$. Together with the successful construction of $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$ for $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, we know that $\mathcal{C}$ will never fail. Here we note that the adversary $\mathcal{A}$ can not obtain $\mathsf{DK}_{\mathsf{ID}^*_{[i]},\mathsf{t}^*}$ for any $i \in [i^*, |\mathsf{ID}^*|]$ since $\mathsf{ID}^*_{[i^*]} \in \mathsf{RL}_{\mathsf{t}^*}$.

Challenge Query. Suppose that $\mathcal{A}$ makes this query on $(\mathsf{ID}^*,\mathsf{t}^*,\mathsf{M}_0,\mathsf{M}_1)$, which satisfies the conditions required for the adaptive-identity security game. Then $\mathcal{C}$ checks whether $\ell^*(:= |\mathsf{ID}^*|) \geqslant i^*$, $\widetilde{\mathsf{ID}^*_{[i^*]}} \in \mathcal{CH}_1$, $\mathsf{ID}^*_{[i^*-1]}\|\mathsf{t}^* \in \mathcal{CH}_2$ holds. If not, $\mathcal{C}$ fails and aborts. Otherwise, $\mathcal{C}$ picks the challenge bit $b \xleftarrow{\$} \{0,1\}$, and runs $\mathbf{Encrypt}(\mathsf{PP},\mathsf{ID}^*,\mathsf{t}^*,\mathsf{M}_b) \to \big(c_0,\ (\mathbf{c}_{i,1},\mathbf{c}_{i,2})_{i\in[\ell^*]},\ \mathbf{c}_{L+1}\big)$, where $c_0, (\mathbf{c}_{i^*,1},\mathbf{c}_{i^*,2})$ are redefined as follows. Recall that $\widehat{\mathbf{v}} = (v_0,v_1,\cdots,v_{2m}) \in \mathbb{Z}_q^{2m+1}$, and we let $\mathbf{v}_1 := (v_1,\cdots,v_m) \in \mathbb{Z}_q^m$, $\mathbf{v}_2 := (v_{m+1},\cdots,v_{2m}) \in \mathbb{Z}_q^m$. After that, $\mathcal{C}$ sets $c_0 \leftarrow v_0 + \mathbf{u}^\top(\sum_{i\in[\ell^*]\backslash\{i^*\}}\mathbf{s}_i) + \mathbf{u}^\top\mathbf{s}_{L+1} + \mathsf{M}_b\lfloor\frac{q}{2}\rfloor$, $\mathbf{c}_{i^*,1} \leftarrow \mathbf{v}_1$, $\mathbf{c}_{i^*,2} \leftarrow \mathbf{v}_2$, where $(\mathbf{s}_i)_{i\in[\ell^*]\backslash\{i^*\}}$ and $\mathbf{s}_{L+1}$ are already selected in the algorithm **Encrypt**. Then $\mathcal{C}$ returns the challenge ciphertext $\mathsf{CT}^* := \big(c_0,\ (\mathbf{c}_{i,1},\mathbf{c}_{i,2})_{i\in[\ell^*]},\ \mathbf{c}_{L+1}\big)$ to $\mathcal{A}$.

Finally, when $\mathcal{A}$ outputs $b' \in \{0,1\}$ as the guess for b at some point, $\mathcal{C}$ checks whether $\mathcal{A}$ has issued a secret key reveal query on $\mathsf{ID}^*_{[i^*]}$. If not, $\mathcal{C}$ fails and aborts. Otherwise, $\mathcal{C}$ outputs 1 in case $b' = b$, and outputs 0 for $b' \neq b$.

Suppose that the event $\mathsf{Success}$ happens, and then we can analyze the challenge query as follows. (1) If the $\mathsf{LWE}_{n,2m+1,q,\chi}$ problem instance $(\widehat{\mathbf{A}},\widehat{\mathbf{v}})$ satisfies $\widehat{\mathbf{v}} = \widehat{\mathbf{A}}^\top\widehat{\mathbf{s}} + \widehat{\mathbf{x}}$ for some $\widehat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_q^n$ and some $\widehat{\mathbf{x}} \hookleftarrow \chi$, we parse $\widehat{\mathbf{x}}$ as $(x_0,x_1,\cdots,x_{2m}) \in \mathbb{Z}^{2m+1}$, and let $\mathbf{x}_1 := (x_1,\cdots,x_m) \in \mathbb{Z}^m$, $\mathbf{x}_2 := (x_{m+1},\cdots,x_{2m}) \in \mathbb{Z}^m$. Then according to $\widehat{\mathbf{v}} = \widehat{\mathbf{A}}^\top\widehat{\mathbf{s}} + \widehat{\mathbf{x}}$ and

$\widehat{\mathbf{A}} = [\ \mathbf{u} \mid \mathbf{A}_0 \mid \mathbf{B}_0\ ]$, we have

$$\begin{aligned} c_0 &= v_0 + \mathbf{u}^\top(\textstyle\sum_{i\in[\ell^*]\setminus\{i^*\}} \mathbf{s}_i) + \mathbf{u}^\top \mathbf{s}_{L+1} + \mathsf{M}_b\lfloor \frac{q}{2} \rfloor \\ &= \mathbf{u}^\top(\textstyle\sum_{i\in[\ell^*]\setminus\{i^*\}} \mathbf{s}_i + \widehat{\mathbf{s}}) + \mathbf{u}^\top \mathbf{s}_{L+1} + x_0 + \mathsf{M}_b\lfloor \frac{q}{2} \rfloor, \\ \mathbf{c}_{i^*,1} &= \mathbf{v}_1 \ = \ \mathbf{A}_0^\top \widehat{\mathbf{s}} + \mathbf{x}_1 \ = \ [\mathbf{A} \cdot (\mathbf{R}^*_{1,i^*} \cdots \mathbf{R}^*_{1,2}\mathbf{R}^*_{1,1})^{-1}]^\top \widehat{\mathbf{s}} + \mathbf{x}_1 \\ &= [\mathbf{A} \cdot \mathbf{P}_1(\widetilde{\mathsf{ID}^*_{[i^*]}})]^\top \widehat{\mathbf{s}} + \mathbf{x}_1, \\ \mathbf{c}_{i^*,2} &= \mathbf{v}_2 \ = \ \mathbf{B}_0^\top \widehat{\mathbf{s}} + \mathbf{x}_2 \ = \ [\mathbf{B} \cdot (\mathbf{R}^*_{2,i^*} \cdots \mathbf{R}^*_{2,2}\mathbf{R}^*_{2,1})^{-1}]^\top \widehat{\mathbf{s}} + \mathbf{x}_2 \\ &= [\mathbf{B} \cdot \mathbf{P}_2(\mathsf{ID}^*_{[i^*-1]} \| \mathsf{t}^*)]^\top \widehat{\mathbf{s}} + \mathbf{x}_2. \end{aligned}$$

Hence the challenge ciphertext CT^* given to $\mathcal{A}$ is a valid encryption of M_b for ID^* on t^*. (2) If the $\mathsf{LWE}_{n,2m+1,q,\chi}$ problem instance $(\widehat{\mathbf{A}}, \widehat{\mathbf{v}})$ is just sampled as $\widehat{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{n\times(2m+1)}$ and $\widehat{\mathbf{v}} \xleftarrow{\$} \mathbb{Z}_q^{2m+1}$, then $(c_0, \mathbf{c}_{i^*,1}, \mathbf{c}_{i^*,2})$ is also uniformly random in $\mathbb{Z}_q^{2m+1}$. Since the distribution of c_0 no longer depends on the value of b, the probability of $\mathcal{A}$ guessing whether $b = 0$ or $b = 1$ is exactly $1/2$, which implies that $\Pr[b' = b] = 1/2$.

Note that once $\mathcal{C}$ fails and aborts at some point in the above game, $\mathcal{C}$ always outputs a uniformly random bit from $\{0, 1\}$. Recall that $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n)$ denotes the advantage of $\mathcal{C}$ for the $\mathsf{LWE}_{n,2m+1,q,\chi}$ problem, and $\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-I}}(n)$ denotes the adaptive-identity security advantage of $\mathcal{A}$ that follows the **Type-I** strategy. Besides, for the case of $\widehat{\mathbf{v}} = \widehat{\mathbf{A}}^\top \widehat{\mathbf{s}} + \widehat{\mathbf{x}}$, we use $\widehat{\mathbf{v}}_1, \mathsf{Success}_1$ to denote $\widehat{\mathbf{v}}, \mathsf{Success}$, respectively. Similarly, for the case of $\widehat{\mathbf{v}} \xleftarrow{\$} \mathbb{Z}_q^{2m+1}$, we use $\widehat{\mathbf{v}}_2, \mathsf{Success}_2$ to denote $\widehat{\mathbf{v}}, \mathsf{Success}$, respectively. Then we have

$$\begin{aligned} \mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n) &= \left| \Pr[\mathcal{C}(\widehat{\mathbf{A}}, \widehat{\mathbf{v}}_1) = 1] - \Pr[\mathcal{C}(\widehat{\mathbf{A}}, \widehat{\mathbf{v}}_2) = 1] \right| \\ &= \left| \Pr[b' = b \mid \mathsf{Success}_1] \cdot \Pr[\mathsf{Success}_1] + \tfrac{1}{2}\Pr[\overline{\mathsf{Success}_1}] - \tfrac{1}{2}\Pr[\mathsf{Success}_2] - \tfrac{1}{2}\Pr[\overline{\mathsf{Success}_2}] \right| \\ &= \left| \Pr[b' = b \mid \mathsf{Success}_1] \cdot \Pr[\mathsf{Success}_1] - \tfrac{1}{2}\Pr[\mathsf{Success}_1] \right| \\ &= \Pr[\mathsf{Success}_1] \cdot \left| \Pr[b' = b \mid \mathsf{Success}_1] - \tfrac{1}{2} \right| \\ &\geqslant \Pr[\mathsf{Success}_1] \cdot [\tfrac{1}{2}\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-I}}(n) - \mathsf{negl}(n)] \\ &= 1/(2L \cdot Q_{\mathbf{H}_1}^{i^*} \cdot Q_{\mathbf{H}_2}) \cdot (1 - \mathsf{negl}(n)) \cdot [\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-I}}(n) - \mathsf{negl}(n)] \\ &\geqslant 1/(2L \cdot Q_{\mathbf{H}_1}^{L} \cdot Q_{\mathbf{H}_2}) \cdot (1 - \mathsf{negl}(n)) \cdot [\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-I}}(n) - \mathsf{negl}(n)]. \end{aligned}$$

Thus we complete the proof of Lemma 5. □

Similarly, we present the following result against an adversary $\mathcal{A}$ that uses the **Type-II** strategy. Its proof proceeds analogously to that of Lemma 5, and thus is given in Appendix C.

Lemma 7. *Suppose that a PPT adversary $\mathcal{A}$ follows the* **Type-II** *strategy, and its adaptive-identity security advantage is denoted by $\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-II}}(n)$. Besides, let $Q_{\mathbf{H}_1}$ denote the maximum numbers of queries made by $\mathcal{A}$ to the random oracle $\mathbf{H}_1$. Then there exits a PPT algorithm $\mathcal{C}$, whose advantage for the $\mathsf{LWE}_{n,2m+1,q,\chi}$ ($\chi = \mathcal{D}_{\mathbb{Z}^{2m+1},\alpha q}$) problem is denoted by $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n)$, such that*

$$\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-II}}(n) \leqslant (2L \cdot Q_{\mathbf{H}_1}^{L+1})\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n) + \mathsf{negl}(n).$$

Actually, the proof of Lemma 7 (in Appendix C) only uses the first $m + 1$ samples of the problem instance of $\mathsf{LWE}_{n,2m+1,q,\chi}$. Namely, we only need to consider the $\mathsf{LWE}_{n,m+1,q,\chi'}$ problem where $\chi' = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$. Therefore, the notation $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n)$ in Lemma 7 can also be replaced by $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}'}(n)$, which denotes the advantage of $\mathcal{C}$ for the $\mathsf{LWE}_{n,m+1,q,\chi'}$ problem.

Finally, according to the "strategy-dividing lemma" introduced in [11,12], Lemma 5 and Lemma 7, we have

$$\begin{aligned} \mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\mathsf{RHIBE\text{-}ad}}(n) &\leqslant \mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-I}}(n) + \mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-II}}(n) \\ &\leqslant (2L \cdot Q_{\mathbf{H}_1}^{L} \cdot Q_{\mathbf{H}_2})\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n) + \mathsf{negl}(n) + (2L \cdot Q_{\mathbf{H}_1}^{L+1})\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n) + \mathsf{negl}(n) \\ &\leqslant 2L \cdot Q_{\mathbf{H}_1}^{L} \cdot (Q_{\mathbf{H}_2} + Q_{\mathbf{H}_1}) \cdot \mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n) + \mathsf{negl}(n). \end{aligned}$$

It is obtained that $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n) = \mathsf{negl}(n)$, assuming the hardness of the problem $\mathsf{LWE}_{n,2m+1,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^{2m+1},\alpha q}$. Since $2L \cdot Q_{\mathbf{H}_1}^{L} \cdot (Q_{\mathbf{H}_2} + Q_{\mathbf{H}_1})$ is polynomial in n, we know that $\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\mathsf{RHIBE\text{-}ad}}(n) \leqslant \mathsf{negl}(n)$, which completes the proof of Theorem 2.

5 Conclusion

In this paper, we present two new RHIBE schemes with DKER from lattices, and thus simplify the construction of RHIBE scheme provided by Katsumata et al. [11]. Our first scheme needs fewer items than that in [11], and the sizes of items are much smaller in our second scheme. The security of these two new schemes are both based on the hardness of the LWE problem, and our second scheme also achieves the adaptive-identity security.

Acknowledgments. The work in this paper is supported by the National Natural Science Foundation of China (Grant Nos. 11531002, 61572026 and 61722213), the Open Foundation of State Key Laboratory of Cryptology, and the program of China Scholarship Council (CSC) (No. 201703170302).

References

1. Agrawal S, Boneh D, Boyen X. Efficient lattice (H) IBE in the standard model. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2010: 553-572.
2. Agrawal S, Boneh D, Boyen X. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2010: 98-115.
3. Ajtai M. Generating hard instances of the short basis problem. International Colloquium on Automata, Languages, and Programming. Springer, Berlin, Heidelberg, 1999: 1-9.
4. Alwen J, Peikert C. Generating shorter bases for hard random lattices. Theory of Computing Systems, 2011, 48(3): 535-553.
5. Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation. Proceedings of the 15th ACM conference on Computer and communications security. ACM, 2008: 417-426.
6. Boneh D, Franklin M. Identity-based encryption from the Weil pairing. Annual international cryptology conference. Springer, Berlin, Heidelberg, 2001: 213-229.
7. Cash D, Hofheinz D, Kiltz E, et al. Bonsai trees, or how to delegate a lattice basis. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2010: 523-552.
8. Chen J, Lim H W, Ling S, et al. Revocable identity-based encryption from lattices. Australasian Conference on Information Security and Privacy. Springer, Berlin, Heidelberg, 2012: 390-403.
9. Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions. Proceedings of the fortieth annual ACM symposium on Theory of computing. ACM, 2008: 197-206.
10. Horwitz J, Lynn B. Toward hierarchical identity-based encryption. International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2002: 466-481.
11. Katsumata S, Matsuda T, Takayasu A. Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance. International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2019: 441-471.
12. Katsumata S, Matsuda T, Takayasu A. Lattice-based Revocable (Hierarchical) IBE with Decryption Key Exposure Resistance. IACR Cryptology ePrint Archive, 2018: 420. https://eprint.iacr.org/2018/420.
13. Katsumata S, Yamada S. Partitioning via non-linear polynomial functions: more compact IBEs from ideal lattices and bilinear maps. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2016: 682-712.
14. Micciancio D, Goldwasser S. Complexity of Lattice Problems: A Cryptographic Perspective. Vol. 671. Springer Science & Business Media, 2002.

15. Micciancio D, Peikert C. Trapdoors for lattices: Simpler, tighter, faster, smaller. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2012: 700-718.
16. Micciancio D, Regev O. Worst-case to average-case reductions based on Gaussian measures. SIAM Journal on Computing, 2007, 37(1): 267-302.
17. Naor D, Naor M, Lotspiech J. Revocation and tracing schemes for stateless receivers. Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 2001: 41-62.
18. Regev O. On lattices, learning with errors, random linear codes, and cryptography. Proceedings of the thirty-seventh annual ACM symposium on Theory of computing. ACM, 2005: 84-93.
19. Seo J H, Emura K. Revocable identity-based encryption revisited: Security model and construction. International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2013: 216-234.
20. Shamir A. Identity-based cryptosystems and signature schemes. Workshop on the theory and application of cryptographic techniques. Springer, Berlin, Heidelberg, 1984: 47-53.
21. Takayasu A, Watanabe Y. Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance. Australasian Conference on Information Security and Privacy. Springer, Cham, 2017: 184-204.

Supplemental Material

Appendix A: Proof of Lemma 3

We provide the proof of Lemma 3 using the following games.

Game 0. This is the real security game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$.

Game 1. In this game, we change the way that the matrices $(\mathbf{C}_j)_{j\in[L+1]}$ in PP are generated for the setup algorithm. At first, $\mathcal{C}$ samples $\mathbf{R}_j^* \xleftarrow{\$} \{-1,1\}^{m\times m}$ for $j \in [L+1]$, and keeps these matrices as a part of $\mathsf{SK}_{\mathsf{KGC}}$. Next, $\mathcal{C}$ sets $(\mathbf{C}_j)_{j\in[L+1]}$ as follows:

$$\mathbf{C}_j := \begin{cases} \mathbf{A}\mathbf{R}_j^* - H(\utilde{\mathsf{id}_j^*})\mathbf{G} & \text{for} \quad j \in [i^*-1], \\ \mathbf{A}\mathbf{R}_j^* - H(\mathsf{id}_j^*)\mathbf{G} & \text{for} \quad j = i^*, \\ \mathbf{A}\mathbf{R}_j^* & \text{for} \quad j \in [i^*+1, L], \\ \mathbf{A}\mathbf{R}_j^* - H(i^*\|\mathsf{t}^*)\mathbf{G} & \text{for} \quad j = L+1. \end{cases}$$

As the proof in [12] (the full version of [11]) shows, the distribution of PP in Game 1 is statistically close to that in Game 0,

Game 2. The changes made in this game are the most important part of our security proof. First of all, we change the way that the matrix $\mathbf{A}$ in PP is generated for the setup algorithm. Instead of running $\mathsf{TrapGen}(1^n, 1^m, q)$ in Game 1, $\mathcal{C}$ selects $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ without the trapdoor $\mathbf{T}_{\mathbf{A}}$ in $\mathsf{SK}_{\mathsf{KGC}}$. The choice of $(\mathbf{C}_j)_{j\in[L+1]}$ remains as in Game 1. Though $\mathcal{C}$ does not own the trapdoor $\mathbf{T}_{\mathbf{A}}$, we still have the following result, whose proof is similar to that in [12] and thus is omitted here. We note that its proof will make heavy use of the algorithms $\mathsf{SampleRight}$, $\mathsf{SampleBasisRight}$, together with the algorithms $\mathsf{SampleLeft}$, $\mathsf{SampleBasisLeft}$.

Lemma 8. *The setup algorithm is changed as above. For any* $\mathsf{CH} \in \left(\mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}\right)^{\leqslant L}$ *with* $|\mathsf{CH}| = \ell$, *and any* $i \in [L]$, $\mathsf{t} \in \mathbb{Z}_q^{n-1}$, $\mathbf{v} \in \mathbb{Z}_q^n$,

(1) if $\mathsf{CH} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}_{[i^*]}^*})$*, then the challenger* $\mathcal{C}$ *is able to construct a short basis* $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{CH})]}$ *distributed statistically close to* $\mathcal{D}_{Basis}(\Lambda_q^{\perp}([\mathbf{A} \mid \mathbf{E}(\mathsf{CH})]), \sigma_{\ell-1})$*, and is also able to construct a short vector* $\mathbf{e}$ *distributed statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{v}}([\mathbf{A}|\mathbf{E}(\mathsf{CH})]),\sigma_{\ell-1}}$*;*

(2) if $\mathsf{CH} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}_{[i^*]}^*})$ *or* $(i, \mathsf{t}) \neq (i^*, \mathsf{t}^*)$*, then the challenger* $\mathcal{C}$ *is able to construct a short basis* $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{CH})|\mathbf{F}(i,\mathsf{t})]}$ *distributed statistically close to* $\mathcal{D}_{Basis}(\Lambda_q^{\perp}([\mathbf{A} \mid \mathbf{E}(\mathsf{CH}) \mid \mathbf{F}(i,\mathsf{t})]), \sigma_\ell)$*, and is also able to construct a short vector* $\mathbf{e}$ *distributed statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{v}}([\mathbf{A}|\mathbf{E}(\mathsf{CH})|\mathbf{F}(i,\mathsf{t})]),\sigma_\ell}$*.*

Besides, we change the way that the vectors $(\mathbf{u}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*),\theta})_{\theta\in\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*)}}$ stored in nodes of $\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*)}$ are generated. When the vector $\mathbf{u}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*),\theta}$ for some $\theta \in \mathsf{BT}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*)}$ must be defined for $\mathcal{C}$ to answer $\mathcal{A}$'s query, $\mathcal{C}$ proceeds as follows. If $\theta \in \mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*)}, \eta_{\mathsf{ID}_{[i^*]}^*})$, $\mathcal{C}$ first samples $\mathbf{e}_{\mathsf{ID}_{[i^*]}^*,\theta} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{(i^*+1)m},\sigma_{i^*-1}}$, and then sets $\mathbf{u}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*),\theta} := [\mathbf{A} \mid \mathbf{E}(\widetilde{\mathsf{ID}_{[i^*]}^*})]\mathbf{e}_{\mathsf{ID}_{[i^*]}^*,\theta}$. If $\theta \in \mathsf{BT}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*)} \backslash \mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*)}, \eta_{\mathsf{ID}_{[i^*]}^*})$, $\mathcal{C}$ first samples $\mathbf{e}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*),\mathsf{t}^*,\theta} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{(i^*+1)m},\sigma_{i^*-1}}$, and then sets $\mathbf{u}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*),\theta} := \mathbf{u} - [\mathbf{A} \mid \mathbf{E}(\mathsf{pa}(\mathsf{ID}_{[i^*]}^*)) \mid \mathbf{F}(i^*, \mathsf{t}^*)]\mathbf{e}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*),\mathsf{t}^*,\theta}$. $\mathcal{C}$ keeps the obtained $\mathbf{e}_{\mathsf{ID}_{[i^*]}^*,\theta}$ or $\mathbf{e}_{\mathsf{pa}(\mathsf{ID}_{[i^*]}^*),\mathsf{t}^*,\theta}$ secret for future use.

With the preparation above, now we prove that the challenger $\mathcal{C}$ is able to answer any allowed queries made by the adversary $\mathcal{A}$ that follows the **Type-I-i^*** strategy for some $i^* \in [\ell^*]$. In the following we summarize the items which $\mathcal{C}$ must construct to respond to $\mathcal{A}$'s queries:

- (a) $\mathsf{SK}_{\mathsf{ID}} = \Big(\mathsf{BT}_{\mathsf{ID}},\ (\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID})},\eta_{\mathsf{ID}})},\ \mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID})]}\Big)$ for $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L}\backslash\mathsf{prefix}(\mathsf{ID}_{[i^*-1]}^*)$;
- (b) $\mathsf{KU}_{\mathsf{ID},\mathsf{t}} = \Big((\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})},\ (\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}\Big)$ for $\mathsf{ID} \in \{\mathsf{KGC}\}\cup(\mathcal{ID})^{\leqslant L-1}$, $\mathsf{t} \in \mathcal{T}$ and $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, where $\ell = |\mathsf{ID}|$;

- (c) $\mathsf{DK}_{\mathsf{ID},\mathsf{t}} = \Big((\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]},\ \mathbf{g}_{\mathsf{ID},\mathsf{t}}\Big)$ for $(\mathsf{ID},\mathsf{t}) \in (\mathcal{ID})^{\leqslant L} \times \mathcal{T} \setminus \{(\mathsf{ID}^*, \mathsf{t}^*)\}$ and $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$, where $\ell = |\mathsf{ID}|$.

$\mathcal{C}$ needs to generate any item in (a) in the secret key generation query and return it to the adversary $\mathcal{A}$ in the secret key reveal query. In the revoke & key update query (and in the secret key generation query, and at the beginning), $\mathcal{C}$ must return the corresponding items in (b) to $\mathcal{A}$. Similarly, $\mathcal{A}$ is allowed to query any item in (c) as a decryption key reveal query.

Firstly, we consider the items in (a). Undoubtedly, $\mathcal{C}$ is able to generate $\mathsf{BT}_{\mathsf{ID}}$ for any $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L}$. For $\mathsf{ID} \neq \mathsf{ID}^*_{[i^*]}$, we have that $\widetilde{\mathsf{ID}} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$ always holds. According to Lemma 8, for any vector $\mathbf{u}_{\mathsf{pa}(\mathsf{ID}),\theta} \in \mathbb{Z}_q^n$, $\mathcal{C}$ is able to construct a short vector distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}_{\mathsf{pa}(\mathsf{ID}),\theta}}([\mathbf{A}|\mathbf{E}(\widetilde{\mathsf{ID}})]),\sigma_{\ell-1}}$, where $\ell = |\widetilde{\mathsf{ID}}| = |\mathsf{ID}|$. Then $\mathcal{C}$ just sets this short vector as $\mathbf{e}_{\mathsf{ID},\theta}$. This shows that $\mathcal{C}$ is able to construct $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID})},\eta_{\mathsf{ID}})}$. For $\mathsf{ID} = \mathsf{ID}^*_{[i^*]}$, $\mathcal{C}$ has already constructed $(\theta, \mathbf{e}_{\mathsf{ID}^*_{[i^*]},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})},\eta_{\mathsf{ID}^*_{[i^*]}})}$ since we change the way that $(\mathbf{u}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]}),\theta})_{\theta\in\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})}}$ are generated. Therefore, $\mathcal{C}$ can construct $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID})},\eta_{\mathsf{ID}})}$ for any $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L}$. Again due to Lemma 8, the condition for $\mathcal{C}$ to construct $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{ID})]}$ is $\mathsf{ID} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$, which is exactly equivalent to $\mathsf{ID} \notin \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]})$. As a conclusion, $\mathcal{C}$ is able to construct $\mathsf{SK}_{\mathsf{ID}}$ for any $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L} \setminus \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]})$. Note that $\mathcal{C}$ can also generate $(\theta, \mathbf{e}_{\mathsf{ID},\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID})},\eta_{\mathsf{ID}})}$ for any $\mathsf{ID} \in \mathsf{prefix}(\mathsf{ID}^*_{[i^*-1]})$, which will play an important role in the construction of $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$ in (b) and (c).

Secondly, let us deal with the items in (b). Suppose that $(\mathsf{ID},\mathsf{t}) \neq (\mathsf{ID}^*_{[i^*-1]}, \mathsf{t}^*)$. If $|\mathsf{ID}| = i^* - 1$ and $\mathsf{t} = \mathsf{t}^*$, then $\mathsf{ID} \neq \mathsf{ID}^*_{[i^*-1]}$ must hold. Thus we have $\mathsf{ID} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$. If $|\mathsf{ID}| \neq i^* - 1$ or $\mathsf{t} \neq \mathsf{t}^*$, then we have $(|\mathsf{ID}| + 1, \mathsf{t}) \neq (i^*, \mathsf{t}^*)$. According to Lemma 8, for any vector $\mathbf{u} - \mathbf{u}_{\mathsf{ID},\theta} \in \mathbb{Z}_q^n$, $\mathcal{C}$ is able to construct a short vector distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}-\mathbf{u}_{\mathsf{ID},\theta}}([\mathbf{A}|\mathbf{E}(\mathsf{ID})|\mathbf{F}(\ell+1,\mathsf{t})]),\sigma_\ell}$, where $\ell = |\mathsf{ID}|$. Then $\mathcal{C}$ just sets this short vector as $\mathbf{e}_{\mathsf{ID},\mathsf{t},\theta}$. This shows that $\mathcal{C}$ is able to construct $(\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})}$. For the case of $(\mathsf{ID},\mathsf{t}) = (\mathsf{ID}^*_{[i^*-1]}, \mathsf{t}^*)$ where $\mathsf{ID}^*_{[i^*-1]} = \mathsf{pa}(\mathsf{ID}^*_{[i^*]})$, $\mathcal{C}$ is able to construct $\mathbf{e}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]}),\mathsf{t}^*,\theta}$ for any $\theta \in \mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})} \setminus \mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})}, \eta_{\mathsf{ID}^*_{[i^*]}})$, since we change the way that $(\mathbf{u}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]}),\theta})_{\theta\in\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})}}$ are generated. By definition of the **Type-I-i^*** strategy, we must have $\mathsf{ID}^*_{[i^*]} \in \mathsf{RL}_{\mathsf{t}^*}$. Therefore, either there is no need to construct $\mathsf{KU}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]}),\mathsf{t}^*}$ (if $\mathsf{pa}(\mathsf{ID}^*_{[i^*]}) \in \mathsf{RL}_{\mathsf{t}^*}$), or we have $\mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})}, \mathsf{RL}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]}),\mathsf{t}^*}) \subset \mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})} \setminus \mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})}, \eta_{\mathsf{ID}^*_{[i^*]}})$. Therefore, $\mathcal{C}$ is able to construct any $(\theta, \mathbf{e}_{\mathsf{ID},\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}},\mathsf{RL}_{\mathsf{ID},\mathsf{t}})}$ in (b). As for $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$ in (b), we note that if $\mathsf{ID}' \notin \mathsf{RL}_{\mathsf{t}}$, $\mathcal{C}$ is able to create any $\mathbf{d}_{\mathsf{ID}',\mathsf{t}}$ from combining $(\theta, \mathbf{e}_{\mathsf{ID}',\theta})_{\theta\in\mathsf{Path}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}')},\eta_{\mathsf{ID}'})}$ and $(\theta, \mathbf{e}_{\mathsf{pa}(\mathsf{ID}'),\mathsf{t},\theta})_{\theta\in\mathsf{KUNode}(\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}')},\mathsf{RL}_{\mathsf{pa}(\mathsf{ID}'),\mathsf{t}})}$, which can be generated by $\mathcal{C}$ as stated above. As a conclusion, $\mathcal{C}$ has the ability to construct $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ for $\mathsf{ID} \in \{\mathsf{KGC}\} \cup (\mathcal{ID})^{\leqslant L-1}$, $\mathsf{t} \in \mathcal{T}$ and $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$.

Thirdly, there remain the items in (c). The method to construct $(\mathbf{d}_{\mathsf{ID}_{[i]},\mathsf{t}})_{i\in[\ell]}$ in (c) is similar to the above. Here we only need to consider $\mathbf{g}_{\mathsf{ID},\mathsf{t}}$. If $|\mathsf{ID}| = i^*$, then $\mathsf{ID} \notin \mathsf{prefix}(\widetilde{\mathsf{ID}^*_{[i^*]}})$. If $|\mathsf{ID}| \neq i^*$, then $(|\mathsf{ID}|, \mathsf{t}) \neq (i^*, \mathsf{t}^*)$. According to Lemma 8, for any vector $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathcal{C}$ is able to construct a short vector distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}([\mathbf{A}|\mathbf{E}(\mathsf{ID})|\mathbf{F}(\ell,\mathsf{t})]),\sigma_\ell}$, where $\ell = |\mathsf{ID}|$. Then $\mathcal{C}$ just sets this short vector as $\mathbf{g}_{\mathsf{ID},\mathsf{t}}$. As a conclusion, $\mathcal{C}$ is able to construct $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ for $(\mathsf{ID},\mathsf{t}) \in (\mathcal{ID})^{\leqslant L} \times \mathcal{T} \setminus \{(\mathsf{ID}^*, \mathsf{t}^*)\}$ and $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{t}}$. Note that $\mathsf{ID}^*_{[i^*]} \in \mathsf{RL}_{\mathsf{t}^*}$, thus we have $\mathsf{ID}^* \in \mathsf{RL}_{\mathsf{t}^*}$, which implies that $\mathbf{d}_{\mathsf{ID}^*,\mathsf{t}^*}$ does not exist.

From the description of the algorithm $\mathsf{TrapGen}$, the two matrices $\mathbf{A}$ in Game 1 and Game 2 are statistically indistinguishable. Moreover, according to Lemma 8 and Lemma 2, the distributions of $\mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{ID},\mathsf{t}}, \mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ provided to the adversary $\mathcal{A}$ in Game 2 are statistically close to those in Game 1, and so are the distributions of the vectors $(\mathbf{u}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]}),\theta})_{\theta\in\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*_{[i^*]})}}$. As a conclusion, the adversary $\mathcal{A}$'s advantage in Game 2 is at most negligibly different from its advantage in Game 1.

Game 3. Recall that in the challenge query, upon a query $(\mathsf{M}_0, \mathsf{M}_1)$ with $|\mathsf{M}_0| = |\mathsf{M}_1|$ from $\mathcal{A}$, $\mathcal{C}$ picks the challenge bit $b \xleftarrow{\$} \{0,1\}$, and runs $\mathbf{Encrypt}(\mathsf{PP}, \mathsf{ID}^*, \mathsf{t}^*, \mathsf{M}_b) \rightarrow \Big(c_0,\ (\mathbf{c}_i)_{i\in[\ell]},\ \mathbf{c}_{L+1}\Big)$. In this game, we reset c_0 and $\mathbf{c}_{i^*}$ as follows. Recall that in the algorithm **Encrypt**, we have already selected $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$ for $i \in [\ell^*] \cup \{L+1\}$, and $x \hookleftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$. Besides, let $\mathcal{C}$ sample $\mathbf{x} \hookleftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$. Then define $w := \mathbf{u}^\top \mathbf{s}_{i^*}, \mathbf{w} := \mathbf{A}^\top \mathbf{s}_{i^*}$, and compute $v := w + x \in \mathbb{Z}_q, \mathbf{v} := \mathbf{w} + \mathbf{x} \in \mathbb{Z}_q^m$. Next, $\mathcal{C}$ sets

$$\begin{cases} c_0 := v + \mathbf{u}^\top(\sum_{i\in[\ell^*]\setminus\{i^*\}} \mathbf{s}_i) + \mathbf{u}^\top \mathbf{s}_{L+1} + \mathsf{M}_b\lfloor \frac{q}{2} \rfloor, \\ \mathbf{c}_{i^*} \leftarrow \mathsf{ReRand}([\mathbf{I}_m \mid \mathbf{R}^*], \mathbf{v}, \alpha q, \frac{\alpha'}{2\alpha}), \quad \text{where} \quad \mathbf{R}^* := [\mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_{i^*}^* \mid \mathbf{R}_{L+1}^*]. \end{cases}$$

Here the algorithm ReRand is introduced in [13] for noise re-randomization. One can also refer to [12] for its definition. Besides, the proof in [12] also shows that the change of $\mathbf{c}_{i^*}$ alters the view of $\mathcal{A}$ only negligibly. In addition, the generation for c_0 is actually unchanged. As a conclusion, Game 2 and Game 3 are statistically indistinguishable.

Game 4. In this game, we further change the way that the challenge ciphertext CT^* is created. Instead of setting $w := \mathbf{u}^\top \mathbf{s}_{i^*}, \mathbf{w} := \mathbf{A}^\top \mathbf{s}_{i^*}$ in Game 3, we let $\mathcal{C}$ sample $w \xleftarrow{\$} \mathbb{Z}_q, \mathbf{w} \xleftarrow{\$} \mathbb{Z}_q^m$. The remainder of Game 4 is the same as Game 3. As the proof in [12] shows, Game 3 and Game 4 are computationally indistinguishable for the PPT adversary $\mathcal{A}$, assuming the hardness of the problem $\mathsf{LWE}_{n,m+1,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$.

In Game 4, according to $w \xleftarrow{\$} \mathbb{Z}_q$ and $c_0 = w + [x + \mathbf{u}^\top(\sum_{i\in[\ell^*]\setminus\{i^*\}} \mathbf{s}_i) + \mathbf{u}^\top \mathbf{s}_{L+1}] + \mathsf{M}_b\lfloor \frac{q}{2} \rfloor$, the probability of $\mathcal{A}$ guessing whether $b = 0$ or $b = 1$ is exactly 1/2. Namely, $\mathcal{A}$'s advantage in Game 4 is zero. According to the analysis for the above games, it is obtained that $\mathcal{A}$'s advantage in Game 0 is negligible, and thus we complete the proof of Lemma 3.

Appendix B: Proof of Lemma 4

The proof of Lemma 4 is similar to that of Lemma 3 in Appendix A. Below we only point out the part different from Appendix A in each game.

In Game 1, the challenger $\mathcal{C}$ sets $(\mathbf{C}_j)_{j\in[L+1]}$ as follows:

$$\mathbf{C}_j := \begin{cases} \mathbf{A}\mathbf{R}_j^* - H(\mathsf{id}_j^*)\mathbf{G} & \text{for} \quad j \in [\ell^*], \\ \mathbf{A}\mathbf{R}_j^* & \text{for} \quad j \in [\ell^*+1, L], \\ \mathbf{A}\mathbf{R}_j^* - H(\ell^* \| \mathsf{t}^*)\mathbf{G} & \text{for} \quad j = L+1. \end{cases}$$

In Game 2, similar to Lemma 8, for any $\mathsf{CH} \in \big(\mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}\big)^{\leqslant L}$ with $|\mathsf{CH}| = \ell$, and any $i \in [L], \mathsf{t} \in \mathbb{Z}_q^{n-1}, \mathbf{v} \in \mathbb{Z}_q^n$, the challenger $\mathcal{C}$ is able to construct a short basis $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{CH})]}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp([\mathbf{A} \mid \mathbf{E}(\mathsf{CH})]), \sigma_{\ell-1})$, and a short vector $\mathbf{e}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{v}}([\mathbf{A}|\mathbf{E}(\mathsf{CH})]),\sigma_{\ell-1}}$, if $\mathsf{CH} \notin \mathsf{prefix}(\mathsf{ID}^*)$; and $\mathcal{C}$ is also able to construct a short basis $\mathbf{T}_{[\mathbf{A}|\mathbf{E}(\mathsf{CH})|\mathbf{F}(i,\mathsf{t})]}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp([\mathbf{A} \mid \mathbf{E}(\mathsf{CH}) \mid \mathbf{F}(i,\mathsf{t})]), \sigma_\ell)$, and a short vector $\mathbf{e}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{v}}([\mathbf{A}|\mathbf{E}(\mathsf{CH})|\mathbf{F}(i,\mathsf{t})]),\sigma_\ell}$, if $\mathsf{CH} \notin \mathsf{prefix}(\mathsf{ID}^*)$ or $(i,\mathsf{t}) \neq (\ell^*, \mathsf{t}^*)$.

Besides, we need to change the way that the vectors $(\mathbf{u}_{\mathsf{pa}(\mathsf{ID}^*),\theta})_{\theta\in\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*)}}$ stored in nodes of $\mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*)}$ are generated. When the vector $\mathbf{u}_{\mathsf{pa}(\mathsf{ID}^*),\theta}$ for some $\theta \in \mathsf{BT}_{\mathsf{pa}(\mathsf{ID}^*)}$ must be defined for $\mathcal{C}$ to answer $\mathcal{A}$'s query, $\mathcal{C}$ first samples $\mathbf{e}_{\mathsf{pa}(\mathsf{ID}^*),\mathsf{t}^*,\theta} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{(\ell^*+1)m},\sigma_{\ell^*-1}}$, and then sets $\mathbf{u}_{\mathsf{pa}(\mathsf{ID}^*),\theta} := \mathbf{u} - [\mathbf{A} \mid \mathbf{E}(\mathsf{pa}(\mathsf{ID}^*)) \mid \mathbf{F}(\ell^*, \mathsf{t}^*)]\mathbf{e}_{\mathsf{pa}(\mathsf{ID}^*),\mathsf{t}^*,\theta}$. $\mathcal{C}$ keeps the obtained $\mathbf{e}_{\mathsf{pa}(\mathsf{ID}^*),\mathsf{t}^*,\theta}$ secret.

Similarly, with the preparation above, we can also prove that the challenger $\mathcal{C}$ is able to answer any allowed queries made by the adversary $\mathcal{A}$ that follows the **Type-II** strategy. Namely, $\mathcal{C}$ is able to construct the following items:

- (a) $\mathsf{SK}_{\mathsf{ID}}$ for $\mathsf{ID} \in (\mathcal{ID})^{\leqslant L} \setminus \mathsf{prefix}(\mathsf{ID}^*)$;

- (b) $\mathsf{KU}_{\mathsf{ID},\mathsf{t}}$ for $\mathsf{ID} \in \{\mathsf{KGC}\} \cup (\mathcal{ID})^{\leqslant L-1}$, $\mathsf{t} \in \mathcal{T}$ and $\mathsf{ID} \notin \mathsf{RL}_\mathsf{t}$;
- (c) $\mathsf{DK}_{\mathsf{ID},\mathsf{t}}$ for $(\mathsf{ID},\mathsf{t}) \in (\mathcal{ID})^{\leqslant L} \times \mathcal{T} \setminus \{(\mathsf{ID}^*,\mathsf{t}^*)\}$ and $\mathsf{ID} \notin \mathsf{RL}_\mathsf{t}$.

In Game 3, $\mathcal{C}$ defines $w := \mathbf{u}^\top \mathbf{s}_{L+1}, \mathbf{w} := \mathbf{A}^\top \mathbf{s}_{L+1}$, and sets

$$\begin{cases} c_0 := v + \mathbf{u}^\top (\sum_{i\in[\ell^*]} \mathbf{s}_i) + \mathsf{M}_b \lfloor \frac{q}{2} \rfloor, \\ \mathbf{c}_{L+1} \leftarrow \mathsf{ReRand}([\mathbf{I}_m \mid \mathbf{R}^*], \mathbf{v}, \alpha q, \frac{\alpha'}{2\alpha}), \quad \text{where} \quad \mathbf{R}^* := [\mathbf{R}_1^* \mid \cdots \mid \mathbf{R}_{\ell^*}^* \mid \mathbf{R}_{L+1}^*]. \end{cases}$$

In Game 4, instead of setting $w := \mathbf{u}^\top \mathbf{s}_{L+1}, \mathbf{w} := \mathbf{A}^\top \mathbf{s}_{L+1}$, $\mathcal{C}$ samples $w \xleftarrow{\$} \mathbb{Z}_q, \mathbf{w} \xleftarrow{\$} \mathbb{Z}_q^m$.

Similarly, for the PPT adversary $\mathcal{A}$, we can prove that any two consecutive games of Games 0, 1, 2 and 3 are statistically indistinguishable, and that Game 3 and Game 4 are computationally indistinguishable, assuming the hardness of the problem $\mathsf{LWE}_{n,m+1,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^{m+1},\alpha q}$. Besides, in Game 4 the probability of $\mathcal{A}$ guessing whether $b = 0$ or $b = 1$ is exactly $1/2$, which implies that $\mathcal{A}$'s advantage is zero in Game 4. Combining everything together, we conclude that $\mathcal{A}$'s advantage in Game 0 (the real security game) is negligible, and thus we complete the proof of Lemma 4.

Appendix C: Proof of Lemma 7

Given the problem instance of $\mathsf{LWE}_{n,2m+1,q,\chi}$ as $(\widehat{\mathbf{A}}, \widehat{\mathbf{v}})$ with $\widehat{\mathbf{A}} = [\ \mathbf{a}_0 \mid \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_{2m}\ ] \in \mathbb{Z}_q^{n\times(2m+1)}$ and $\widehat{\mathbf{v}} = (v_0, v_1, \cdots, v_{2m}) \in \mathbb{Z}_q^{2m+1}$, the algorithm $\mathcal{C}$ sets $\mathbf{u} := \mathbf{a}_0 \in \mathbb{Z}_q^n$, $\mathbf{A}_0 := [\ \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_m\ ] \in \mathbb{Z}_q^{n\times m}$ such that $\widehat{\mathbf{A}} = [\ \mathbf{u} \mid \mathbf{A}_0 \mid \cdots\ ]$. Besides, $\mathcal{C}$ selects $\ell^* \xleftarrow{\$} [L]$ as the guess for the length of the challenge identity ID^*, and samples $Q_{1,j}^* \xleftarrow{\$} [Q_{\mathbf{H}_1}]$, $\mathbf{R}_{1,j}^* \hookleftarrow \mathcal{D}_{m\times m}$ for $j \in [\ell^* + 1]$. Then $\mathcal{C}$ sets $\mathbf{A} := \mathbf{A}_0(\mathbf{R}_{1,\ell^*+1}^* \cdots \mathbf{R}_{1,2}^* \mathbf{R}_{1,1}^*)$, and runs $(\mathbf{B}, \mathbf{T}_\mathbf{B}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. Finally, $\mathcal{C}$ publishes the public parameters $\mathsf{PP} := (\mathbf{A}, \mathbf{B}, \mathbf{u})$.

The random oracle $\mathbf{H}_1$ is operated just as that in the proof of Lemma 5, where the parameter i^* should be replaced by $\ell^* + 1$. Then similar to Lemma 6, for $\mathsf{CH} \in (\{0,1,2\}^\omega)^{\leqslant L+1}$ with $|\mathsf{CH}| = \ell$, we can prove that the algorithm $\mathcal{C}$ is able to construct a short basis $\mathbf{T}_{\mathbf{A}\cdot\mathbf{P}_1(\mathsf{CH})}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp(\mathbf{A} \cdot \mathbf{P}_1(\mathsf{CH})), \sigma_{\ell-1})$, if $\mathsf{CH} \notin \mathcal{CH}_1 := \big\{\mathsf{CH} \in (\{0,1,2\}^\omega)^{\leqslant L+1} \mid |\mathsf{CH}| \leqslant \ell^* + 1,$ and $\mathbf{H}_1(\mathsf{CH}_{[j]}) = \mathbf{R}_{1,j}^*$ for $j = 1, 2, \cdots, |\mathsf{CH}|\big\}$. As for the random oracle $\mathbf{H}_2$, once $\mathcal{A}$ queries $\mathbf{H}_2$ on some $\mathsf{CH} \in (\{0,1,2\}^\omega)^{\leqslant L}$, $\mathcal{C}$ just selects $\mathbf{R} \hookleftarrow \mathcal{D}_{m\times m}$ and then returns $\mathbf{H}_2(\mathsf{CH}) := \mathbf{R}$ to $\mathcal{A}$. Note that $\mathcal{C}$ owns the trapdoor $\mathbf{T}_\mathbf{B}$. As a consequence, for any $\mathsf{CH} \in (\{0,1,2\}^\omega)^{\leqslant L}$ with $|\mathsf{CH}| = \ell$, the algorithm $\mathcal{C}$ is able to construct a short basis $\mathbf{T}_{\mathbf{B}\cdot\mathbf{P}_2(\mathsf{CH})}$ distributed statistically close to $\mathcal{D}_{Basis}(\Lambda_q^\perp(\mathbf{B} \cdot \mathbf{P}_2(\mathsf{CH})), \sigma_{\ell-1})$.

Set ID^*, t^* as the challenge identity and time period, and let $\mathsf{Success}$ be the event that $|\mathsf{ID}^*| = \ell^*$, $\mathsf{ID}^* \| \mathsf{t}^* \in \mathcal{CH}_1$ holds, and $\mathcal{C}$ does not fail due to collisions on $\mathbf{H}_1$ found by $\mathcal{A}$. Then we obtain $\Pr[\mathsf{Success}] = 1/L \cdot 1/Q_{\mathbf{H}_1}^{\ell^*+1} \cdot (1 - \mathsf{negl}(n))$. Similar to the proof of Lemma 5, we can prove that if $\mathsf{Success}$ happens, the algorithm $\mathcal{C}$ will successfully simulate the attack environment for $\mathcal{A}$, and otherwise, $\mathcal{C}$ will fail and abort at some point. Note that in the proof of Lemma 7, $\mathcal{C}$ does not need to deal with BT_ID differently for any ID. When $\mathcal{A}$ makes the challenge query on $(\mathsf{ID}^*, \mathsf{t}^*, \mathsf{M}_0, \mathsf{M}_1)$, if $|\mathsf{ID}^*| = \ell^*$, $\mathsf{ID}^* \| \mathsf{t}^* \in \mathcal{CH}_1$ holds, $\mathcal{C}$ picks the challenge bit $b \xleftarrow{\$} \{0,1\}$, and runs $\mathbf{Encrypt}(\mathsf{PP}, \mathsf{ID}^*, \mathsf{t}^*, \mathsf{M}_b) \to \big(c_0,\ (\mathbf{c}_{i,1}, \mathbf{c}_{i,2})_{i\in[\ell^*]},\ \mathbf{c}_{L+1}\big)$, where $c_0, \mathbf{c}_{L+1}$ are redefined as follows. Recall that $\widehat{\mathbf{v}} = (v_0, v_1, \cdots, v_{2m}) \in \mathbb{Z}_q^{2m+1}$, and we let $\mathbf{v}_1 := (v_1, \cdots, v_m) \in \mathbb{Z}_q^m$. After that, $\mathcal{C}$ sets $c_0 \leftarrow v_0 + \mathbf{u}^\top (\sum_{i\in[\ell^*]} \mathbf{s}_i) + \mathsf{M}_b \lfloor \frac{q}{2} \rfloor$, $\mathbf{c}_{L+1} \leftarrow \mathbf{v}_1$. Then $\mathcal{C}$ returns the challenge ciphertext $\mathsf{CT}^* := \big(c_0,\ (\mathbf{c}_{i,1}, \mathbf{c}_{i,2})_{i\in[\ell^*]},\ \mathbf{c}_{L+1}\big)$ to $\mathcal{A}$.

When $\mathcal{A}$ outputs $b' \in \{0,1\}$ as the guess for b at some point, $\mathcal{C}$ outputs 1 in case $b' = b$, and outputs 0 for $b' \neq b$. The remaining analysis for the relation between $\mathsf{Adv}_{\mathbf{\Pi}_2,L,\mathcal{A}}^{\textbf{Type-II}}(n)$ and $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{LWE}}(n)$ is almost the same as that in the proof of Lemma 5, and finally we can complete the proof of Lemma 7.