Adaptive-ID Secure Revocable Identity-Based Encryption from Lattices via Subset Difference Method

Shantian Cheng and Juanyang Zhang

Division of Mathematical Sciences,
School of Physical and Mathematical Sciences,
Nanyang Technological University
SPMS-MAS-04-20, 21 Nanyang Link, 637371, Singapore
{scheng002,zh0078ng}@ntu.edu.sg

Abstract. In view of the expiration or reveal of user's private credential (or private key) in a realistic scenario, identity-based encryption (IBE) schemes with an efficient key revocation mechanism, or for short, revocable identity-based encryption (RIBE) schemes, become prominently significant. In this paper, we present an RIBE scheme from lattices by combining two Agrawal et al.'s IBE schemes with the subset difference (SD) method. Our scheme is secure against adaptive identity-time attacks in the standard model under the learning with errors (LWE) assumption. In particular, our scheme serves as one solution to the challenge posed by Chen et al. (ACISP '12).

1 Introduction

Since Shamir [25] proposed the concept of identity-based encryption (IBE), in which the sender may use the recipient's identity as a public key to encrypt a message, it was a longstanding open problem to construct a practical IBE scheme. Later, Boneh and Franklin [9] proposed the first practical IBE scheme using pairings. After then, IBE becomes extensively attractive from both theoretical and practical perspectives. As a result, there have been many IBE schemes proposed based on various tools and assumptions such as quadratic residue modulo a hard-to-factor integer [12, 10], and lattices [15, 1, 3] as well as pairings [8, 27, 28].

In the practical use of IBE, a user's private key may be expired or revealed for various reasons. In such situations, an efficient key revocation mechanism is needed to keep the whole system stable and secure. To handle this issue, there have been various solutions presented based on pairings. The first revocable IBE (RIBE) scheme was proposed by Boneh and Franklin [9], but their suggestion is inefficient in terms of the workload of the key authority. Later, Boldyreva et al. [7] presented the first selective-ID secure scalable RIBE scheme by combining a fuzzy IBE scheme [22] and the complete subtree (CS) method [20]. The first adaptive-ID secure scalable RIBE scheme was proposed by Libert and Vergnaud [17].

Subsequently, Seo and Emura [24] presented an enhanced security model of RIBE by considering one special realistic threat, which is called decryption key exposure attack, and provided a concrete construction based on pairings. Recently, Lee et al. [16] succeeded in constructing an adaptively secure RIBE scheme based on pairings using the subset difference (SD) method.

However, as a counterpart of pairings, lattices have not been extensively used to construct RIBE schemes. Lattice-based cryptography is widely believed to be resistant against quantum computers. In addition, lattice-based cryptography owns provable security under worst-case hardness assumptions [5, 21, 15]. Therefore, lattices have become a promising and powerful tool to construct a mass of cryptographic primitives. However, to the best of our knowledge, in terms of RIBE schemes, Chen et al. [11] solely offered a selective-ID secure RIBE scheme from lattices. Besides, they left the problem of constructing adaptive-ID secure RIBE schemes from lattices as a challenge.

1.1 Our Results

We present the first adaptive-ID secure RIBE scheme from lattices via the SD method in the standard model. To this end, we employ two adaptive-ID secure IBE schemes for an identity and a time, respectively, proposed by Agrawal et al. [1]. Then, we combine them with the SD method for the revocation and use a secret sharing scheme utilized in fuzzy IBE proposed by Agrawal et al. [4] to share one common random vector in the SD method.

We demonstrate that our construction achieves adaptive-ID security in the standard model under the LWE assumption. Our security proof follows the Libert and Vergnaud's security model [17] with a slight modification that uses a game that captures a strong privacy property, called *indistinguishable from random*, as in [1]. As a result, we provide one solution to the challenge posed by Chen et al. [11].

1.2 Our Techniques

Our new construction of RIBE from lattices consists of the following building blocks:

(i) adaptive-ID secure IBE proposed by Agrawal et al. [1];
(ii) the SD method proposed by [20];
(iii) Shamir's secret sharing scheme utilized in fuzzy IBE proposed by Agrawal et al. [4].

Agrawal et al.'s adaptive-ID secure IBE scheme is a lattice analog of Waters' fully secure IBE scheme [28]. Note that as the time space in RIBE is polynomial size in public parameters, we may just use Agrawal et al.'s selective-ID secure IBE scheme for time part. However, for simplicity and less clutter, we apply two instances of Agrawal et al.'s adaptive-ID secure IBE scheme regarding to users' identities and times, respectively. Our scheme does not just roughly combine the two adaptive-ID secure IBE instances to achieve the adaptive-ID security of RIBE. In contrast to single adaptive-ID secure IBE scheme, the adversary in our RIBE scheme may have already queried the private key for the challenge identity ID^* and the update key for the challenge time T^* before the challenge phase. So we can not easily adopt the non-abort probability as in [1]. Instead, to achieve the adaptive-ID security, we further investigate the abort-resistant hash functions used in [27, 1] and provide more general analysis of the abort-resistant properties of Waters' hash families [27] with respect to our requirements. Then we carefully design the abort events in the series of security challenging games and prove that they are indistinguishable under the properties of the abort-resistant hash functions and the learning with errors (LWE) assumption. Note that the LWE problem is as hard as the worst-case approximation of the short vectors on arbitrary lattices [21].

For revocation, we adopt the SD scheme, an alternative subset cover framework to the well-known CS scheme. Both instantiations are binary tree based and introduced by Naor et al. [20]. However, the main advantage of the SD method is that for r revoked leaf nodes out of total N leaf nodes in one binary tree, the size of the covering set is at most $2r - 1$ (in the worst case, or $1.25r$ in the average case) and independent of N, whereas the covering set in the CS method has size $r \log(N/r)$, logarithmic complexity in N. The tradeoff is that the size of private sets is slightly increased from $O(\log N)$ to $O(\log^2 N)$. While the private key is a long-term key transferred only once via a secure channel and the update key is a short-term key that will be broadcast periodically via public channels, therefore, using the SD method can efficiently reduce the overheads of the key authority.

We require a random n-vector $\mathbf{u}$ to be part of the public parameters. It acts as the secret for sharing and the subsets in one binary tree formed by the same ancestor and same depth descendants act as sharing parties. Different from the CS framework used in prior works [11, 24], the SD method requires a pair of certain subsets, instead of one common node, to determine the non-revocation and generate the decryption key. Thus we can not apply Chen et al.'s methods that directly and randomly split the vector $\mathbf{u}$ inside one node, which indeed is a 2-out-of-2 secret sharing scheme. Based on the SD framework, we have to share $\mathbf{u}$ inside families of subsets formed by the same ancestor and same depth descendants. The size of one such family is a power of 2. Thus it is necessary to consider 2-out-of-n secret sharing scheme. One natural candidate to solve this problem is Shamir's secret sharing scheme. However, to ensure the correctness of dual-Regev type encryption scheme in our scheme, we have to bound the size of vectors in the decryption key. Then we apply the "clear the denominators" of Lagrangian coefficients methods from [4] to revise the encryption and set proper parameters to ensure the correctness.

1.3 Comparison to Chen et al.'s Scheme

Chen et al.'s RIBE scheme [11] is a pioneering work on constructing RIBE schemes from lattices. In Table 1, we elucidate the pros and cons of our scheme in comparison with Chen et al.'s scheme.

	public key size	private key size	update key size	ciphertext size	Security
Chen et al.'s [11]	$\widetilde{O}\left(n^{2+\delta}\right)$	$O\left(\log N\right) \cdot \widetilde{O}\left(n^{1+\delta}\right)$	$r \log \frac{N}{r} \cdot \widetilde{O}\left(n^{1+\delta}\right)$	$\widetilde{O}\left(n^{1+\delta}\right)$	Selective-ID
Ours	$\widetilde{O}\left(n^{2+\delta+\epsilon}\right)$	$O\left(\log^2 N\right) \cdot \widetilde{O}\left(n^{1+\delta}\right)$	$(2r-1) \cdot \widetilde{O}\left(n^{1+\delta}\right)$	$\widetilde{O}\left(n^{1+\delta+\epsilon}\right)$	Adaptive-ID

Table 1. δ is a small constant such that $n^{\delta} > O(\log n)$, and ϵ is a small constant such that $\epsilon < 1/2$. r is the number of revoked users out of N users. When r is small compared with N, our scheme has shorter update key. Both schemes are secure under LWE assumption in the standard model. However, our scheme is based on the LWE assumption with sub-exponential parameters, in contrast to polynomial parameters in Chen et al.'s scheme.

2 Preliminaries

Notation Let $\lambda \in \mathbb{N}$ be the security parameter and 1^{λ} its unary representation. We say a function $d : \mathbb{N} \to \mathbb{R}$ is negligible, if for sufficient large λ, $|d(\lambda)|$ is smaller than reciprocal of any polynomial in λ. For a positive integer a, $[a]$ indicates the set of positive integers less than or equal to a.

2.1 Definition of RIBE Scheme

We recall the definitions for RIBE as defined in [7, 11, 24].

Definition 1 (Revocable IBE). *An identity-based encryption with efficient revocation or simply revocable IBE scheme has seven probabilistic polynomial-time (PPT) algorithms,* **Setup**, **PriKeyGen**, **UpdateKey**, **DecKeyGen**, **Encrypt**, **Decrypt**, *and* **Revoke** *with associated message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, identity space $\mathcal{I}$, and time space $\mathcal{T}$. The size of $\mathcal{T}$ is polynomial in the security parameter. We treat time as discrete as opposed to continuous. There are three kinds of parties in this scheme–key authority, sender and receiver. Each algorithm is run by either one of them. The key authority sustains a revocation list RL and a state st. An algorithm is called stateful if it updates RL or st. The RIBE scheme proceeds as follows:*

Setup$(1^{\lambda}, 1^{N})$ *takes as input a security parameter λ and a maximal number of users N. It outputs public parameters pp, a master key msk, a revocation list RL (initially empty), and a state st. (This is stateful and run by the key authority.)*

PriKeyGen$(\mathsf{pp}, \mathsf{msk}, \mathsf{id}, \mathsf{st})$ *takes as input the public parameters pp, the master key msk, an identity $\mathsf{id} \in \mathcal{I}$, and the state st. It outputs a private key $\mathsf{sk}_{\mathsf{id}}$ and an updated state st. (This is stateful and run by the key authority.)*

UpdateKey$(\mathsf{pp}, \mathsf{msk}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$ *takes as input the public parameters pp, the master key msk, a key update time $\mathsf{t} \in \mathcal{T}$, the revocation list RL at time t, and the state st. It outputs an update key uk_{t}. (This is run by the key authority.)*

DecKeyGen$(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, \mathsf{uk}_{\mathsf{t}})$ *takes as input the public parameters pp, a private key $\mathsf{sk}_{\mathsf{id}}$, and an update key uk_{t}. It outputs a decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$, or $\perp$ if id has been revoked before t. (This is deterministic and run by the receiver.)*

Encrypt$(\mathsf{pp}, \mathsf{id}, \mathsf{t}, b)$ *takes as input the public parameters pp, an identity $\mathsf{id} \in \mathcal{I}$, a time $\mathsf{t} \in \mathcal{T}$, and a message $b \in \mathcal{M}$. It outputs a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$. (This is run by the sender.)*

Decrypt$(\mathsf{pp}, \mathsf{dk}_{\mathsf{id},\mathsf{t}}, \mathsf{ct}_{\mathsf{id},\mathsf{t}})$ *takes as input the public parameters pp, a decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$, and a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$. It outputs a message $b \in \mathcal{M}$ or $\perp$. (This is deterministic and run by the receiver.)*

Revoke$(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$ *takes as input an identity $\mathsf{id} \in \mathcal{I}$ to be revoked, a revocation time $\mathsf{t} \in \mathcal{T}$, the current revocation list RL and a state st. It outputs an updated revocation list RL. (This is stateful and run by the key authority.)*

The correctness condition requires that for any $\lambda \in \mathbb{N}$ and N, any $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathbf{Setup}(1^{\lambda}, 1^{N})$, any $b \in \mathcal{M}$, all possible state st, and a revocation list RL, if $\mathsf{id} \in \mathcal{I}$ is not revoked on a time $\mathsf{t} \in \mathcal{T}$, then for

$$
\begin{aligned}
(\mathsf{sk}_{\mathsf{id}}, \mathsf{st}) &\leftarrow \mathbf{PriKeyGen}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}, \mathsf{st}),\\
\mathsf{uk}_{\mathsf{t}} &\leftarrow \mathbf{UpdatKey}(\mathsf{pp}, \mathsf{msk}, \mathsf{t}, \mathsf{RL}, \mathsf{st}),\\
\mathsf{dk}_{\mathsf{id},\mathsf{t}} &\leftarrow \mathbf{DecKeyGen}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, \mathsf{uk}_{\mathsf{t}}),
\end{aligned}
$$

the following always holds except for a negligible probability:

$$\mathbf{Decrypt}\left(\mathsf{pp}, \mathsf{dk}_{\mathsf{id},\mathsf{t}}, \mathbf{Encrypt}\left(\mathsf{pp}, \mathsf{id}, \mathsf{t}, b\right)\right) = b.$$

The security model of RIBE was introduced by Boldyreva et al. [7], where a selectively secure RIBE was given. Later Libert and Vergnaud [17] proposed a RIBE scheme satisfying adaptive security model. Recently, Seo and Emura [24] advanced the security model by considering the decryption key exposure attack. In this paper, we follow Libert and Vergnaud's security model.

The slight difference is that we formalize the RIBE adaptive security using a game that captures a strong privacy property called *indistinguishable from random*. Agrawal et al. [1] defined this kind of game for IBE. They also referred that indistinguishability from random implies both semantic security and recipient anonymity, and also ciphertext hides the public parameters (pp) used to create it. For RIBE, the game is designed as follows.

Setup: The challenger $\mathcal{B}$ runs $\mathsf{Setup}(1^\lambda, 1^N)$ to generate public parameters pp, a master key msk, a revocation list RL (initially empty), and a state st. Then $\mathcal{B}$ gives pp to the adversary $\mathcal{A}$.

Query: $\mathcal{A}$ is allowed to adaptively make a polynomial number of queries of the following oracles (the oracles share the state) with some restrictions.

1. **PriKeyGen**$(\cdot)$: On input an identity id, return a private key $\mathsf{sk}_{\mathsf{id}}$ by running **PriKeyGen**$(\mathsf{pp}, \mathsf{msk}, \mathsf{id}, \mathsf{st}) \rightarrow \mathsf{sk}_{\mathsf{id}}$.
2. **UpdateKey**$(\cdot)$: On input a time t, return an update key uk_{t} by running **UpdateKey**$(\mathsf{pp}, \mathsf{msk}, \mathsf{t}, \mathsf{RL}, \mathsf{st}) \rightarrow \mathsf{uk}_{\mathsf{t}}$.
3. **Revoke**$(\cdot,\cdot)$: On input an identity id and a time t, update RL by running **Revoke**$(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$.

Challenge: $\mathcal{A}$ outputs an identity id^*, a time t^*, and a plaintext $b^* \in \mathcal{M}$, on which it wishes to be challenged. $\mathcal{B}$ picks a random bit $r \in \{0,1\}$ and a random ciphertext $C \in \mathcal{C}$. If $r = 1$, it sets the challenge ciphertext to $C^* := C$. If $r = 0$, it sets the challenge ciphertext to $C^* :=$ **Encrypt**$(\mathsf{pp}, \mathsf{id}^*, \mathsf{t}^*, b^*)$. It sends C^* as the challenge ciphertext to the adversary $\mathcal{A}$.

Guess: $\mathcal{A}$ may further make a polynomial number of queries of the oracles as in the query phase, then it outputs a bit r'. We say $\mathcal{A}$ succeeds if $r' = r$.

In the above game, the following restrictions must hold:

1. **UpdateKey**$(\cdot)$ and **Revoke**$(\cdot,\cdot)$ can only be queried on time that is greater than or equal to the time of all previous queries.
2. **Revoke**$(\cdot,\cdot)$ can not be queried on time t if **UpdateKey**$(\cdot)$ has already been queried on time t.
3. If **PriKeyGen**$(\cdot)$ was queried on id^*, then **Rovoke**$(\cdot,\cdot)$ must be queried on $(\mathsf{id}^*, \mathsf{t})$ for some $\mathsf{t} \leq \mathsf{t}^*$.

We call such an adversary $\mathcal{A}$ as an $\mathsf{INDr\text{-}RID\text{-}CPA}$ PPT adversary. The advantage of $\mathcal{A}$ attacking a RIBE scheme $\mathcal{RIBE}$ is defined as

$$\mathsf{Adv}^{\mathsf{INDr\text{-}RID\text{-}CPA}}_{\mathcal{RIBE},\mathcal{A}}(\lambda) = \left|\Pr\left[r' = r\right] - \frac{1}{2}\right|.$$

We say that a RIBE scheme $\mathcal{RIBE}$ is indistinguishable from random under adaptive identity-time attacks, if for all $\mathsf{INDr\text{-}RID\text{-}CPA}$ PPT adversary $\mathcal{A}$, we have that its advantage $\mathsf{Adv}^{\mathsf{INDr\text{-}RID\text{-}CPA}}_{\mathcal{RIBE},\mathcal{A}}(\lambda)$ is a negligible function in λ.

2.2 Subset Difference Method

Naor et al. [20] gave two subset-cover revocation schemes with different performances: complete subtree method and subset difference method. The key characteristic of the SD method, which essentially leads to the reduction in the size of covering set, is that in this method any leaf node belongs to substantially more subsets than in the CS method ($O(N)$ instead $\log N$). Furthermore, SD is r-flexible, namely it works with any number of revocations.

Let BT be a full binary tree with N_{max} leaf nodes. Label the nodes in BT by 1 to $2N_{\mathsf{max}} - 1$ in the way that root is labeled 1, if parent is labeled i, then the left child is labeled $2i$ and the right child is labeled

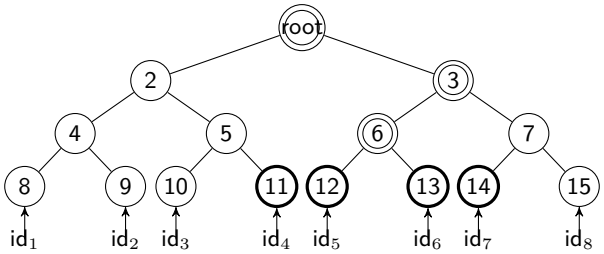

Fig. 1. $R = \{\mathsf{id}_4, \mathsf{id}_5, \mathsf{id}_6, \mathsf{id}_7\}$. Let v_i denote the node labeled i for $i \in [2N_{\mathsf{max}} - 1]$, while the root is labeled 1. First choose v_6 as lca of v_{12} and v_{13}. Get v_6 as a leaf; then v_3 is lca of v_{14} and v_6. Get v_3 as a leaf, $CV_R = \{S_{7,14}\}$; then the root is lca of v_{11} and v_3, $CV_R = \{S_{7,14}, S_{2,11}\}$. Get the root as a leaf. Then, $PV_{\mathsf{id}_2} = \{S_{1,2}, S_{1,4}, S_{1,9}, S_{2,4}, S_{2,9}, S_{4,9}\}$ and $\mathsf{SD.Match}(CV_R, PV_{\mathsf{id}_2})$ outputs $(S_{2,11}, S_{2,9})$. Similarly, $\mathsf{SD.Match}(CV_R, PV_{\mathsf{id}_8})$ outputs $(S_{7,14}, S_{7,15})$. For $\nu \in R$, $\mathsf{SD.Match}(CV_R, PV_\nu)$ outputs $\perp$.

$2i + 1$. One example of the labeling is in Figure 1. Let v_i denote the node labeled i for $i \in [2N_{\mathsf{max}} - 1]$. The depth of a node v_i is the length of the path from the root to v_i. For any $v_i \in \mathsf{BT}$, T_i is defined as a subtree that is rooted at v_i. For any $v_i, v_j \in \mathsf{BT}$, $T_{i,j}$ is defined as a subtree $T_i - T_j$. S_i, $S_{i,j}$ are the sets of leaf nodes of T_i, $T_{i,j}$, respectively.

For a binary tree BT and a subset R of nodes, $ST(R)$ is the Steiner tree induced by R and the root node, that is, the minimal subtree of BT that connects all the leaf nodes in R and the root node.

The SD scheme is summarized as follows:

$\mathsf{SD.Setup}(N_{\mathsf{max}})$: This algorithm takes as input the maximum number of users N_{max} (for simplicity, $N_{\mathsf{max}} = 2^d$). It sets a full binary tree BT of depth d. Every user is assigned to a different leaf node in BT. The collection S is the set of all subsets $S_{i,j}$ with $v_i, v_j \in \mathsf{BT}$ where v_i is an ancestor of v_j. It outputs BT.

$\mathsf{SD.Assign}(\mathsf{BT}, \nu)$: This algorithm takes as input the full binary tree BT and a user ν. Let v_ν be the leaf node assigned to ν and $Path(v_\nu) = (v_{k_0}, v_{k_1}, ..., v_{k_n})$ be the path from the root node v_{k_0} to the leaf node $v_{k_n} = v_\nu$. For all $i, j \in \{k_0, k_1, ..., k_n\}$ such that v_j is a descendant of v_i, it adds $S_{i,j}$ defined by two nodes v_i and v_j in the path into a private set PV_ν. Finally, it outputs $PV_\nu = \{S_{i,j}\}$.

$\mathsf{SD.Cover}(\mathsf{BT}, R)$: This algorithm takes as input the binary tree BT and a revoked set R of users. Let T be the Steiner tree $ST(R)$, then it computes CV_R iteratively by removing nodes from T until T only has a single node as follows (an example is given in Figure 1):

1. Find two leaves v_i and v_j in T such that the least-common-ancestor (lca) v of v_i and v_j does not contain any other leaf node of T in its subtree. Let v_l and v_k be the two children of v such that v_i is a descendant of v_l, and v_j a descendant of v_k. (If there is only one leaf node, make $v_i = v_j$ the leaf, v the root of T and $v_l = v_k = v$.)
2. If $v_l \neq v_i$, $CV_R = CV_R \cup \{S_{l,i}\}$; if $v_k \neq v_j$, $CV_R = CV_R \cup \{S_{k,j}\}$.
3. Remove from T all the descendants of v and make it a leaf.

$\mathsf{SD.Match}(CV_R, PV_\nu)$: This algorithm takes as input a covering set $CV_R = \{S_{i,j}\}$ and a private set $PV_\nu = \{S'_{i',j'}\}$. If it finds two subsets $S_{i,j} \in CV_R$ and $S'_{i',j'} \in PV_\nu$ with $(i = i') \wedge (d_j = d_{j'}) \wedge (j \neq j')$ where d_j is the depth of v_j, then it outputs $(S_{i,j}, S'_{i',j'})$. Otherwise, it outputs $\perp$.

The correctness of the SD scheme is defined as follows: For all $N = 2^d$, $\mathsf{BT} \leftarrow \mathsf{SD.Setup}(N)$, $PV_\nu \leftarrow \mathsf{SD.Assign}(\mathsf{BT}, \nu)$, R a set of leaf nodes in BT, let $CV_R \leftarrow \mathsf{SD.Cover}(\mathsf{BT}, R)$, then

1. If $u \notin R$, then $\mathsf{SD.Match}(CV_R, PV_\nu) = (S_{i,j}, S'_{i',j'})$;
2. If $u \in R$, then $\mathsf{SD.Match}(CV_R, PV_\nu) = \perp$.

Naor et al. [20] characterized the size of a private set and a covering set in the SD method in the following lemma:

Lemma 1. *Let N_{max} be the number of leaf nodes and r be the size of a revoked set. The size of a private set is $O(\log^2 N_{\mathsf{max}})$ and the size of a covering set is at most $2r - 1$ in the SD scheme.*

2.3 Background on Lattices

In this subsection, we review the required knowledge of lattices.

Integer Lattices

Definition 2. *Let $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_k\} \subseteq \mathbb{R}^m$ consist of k linearly independent vectors, where $k \leq m$. Define k-dimensional lattice Λ generated by $\mathbf{B}$ as*

$$\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bc} = \sum_{i \in [k]} c_i \cdot \mathbf{b}_i : \mathbf{c} \in \mathbb{Z}^k\}.$$

For a set of vectors $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_k\} \subseteq \mathbb{R}^m$, we call the norm of $\mathbf{S}$ as $\|\mathbf{S}\| = \max_{1 \leq i \leq k} \|\mathbf{s}_i\|$, where $\|\mathbf{s}\|$ denotes the ℓ_2-norm of the column vector $\mathbf{s}$. Moreover, $\widetilde{\mathbf{S}} := \{\tilde{\mathbf{s}}_1, \cdots, \tilde{\mathbf{s}}_k\} \subseteq \mathbb{R}^m$ represents the Gram-Schmidt orthogonalization of the vectors $\mathbf{s}_1, \cdots, \mathbf{s}_k$ taken in the same order. Let $\|\widetilde{\mathbf{S}}\|$ denote the Gram-Schmidt norm of $\mathbf{S}$.

In our case, we only consider integer lattices, that is, $\Lambda \subseteq \mathbb{Z}^m$.

Definition 3. *For a prime q, a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, define*

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{Ae} = \mathbf{0} \mod q\},$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{Ae} = \mathbf{u} \mod q\}.$$

Trapdoor for Lattices Ajtai [5], Alwen and Peikert [6], Micciancio and Peikert [18] provided methods to produce a matrix $\mathbf{A}$ statistically close to uniform in $\mathbb{Z}_q^{n \times m}$ along with a short basis $\mathbf{T}$ of lattice $\Lambda_q^{\perp}(\mathbf{A})$. We summarize it in the following proposition.

Proposition 1. *Let $q \geq 3$ be a prime and $m \geq 2n \log q$. Then there exists a probabilistic polynomial-time algorithm $\mathsf{TrapGen}(q, n)$ that outputs a pair $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ such that $\mathbf{A}$ is statistically close to uniform and $\mathbf{T_A}$ is a basis for $\Lambda_q^{\perp}(\mathbf{A})$ with length $L = \|\widetilde{\mathbf{T_A}}\| \leq O(\sqrt{m})$ with all but $n^{-\omega(1)}$ probability.*

Discrete Gaussians

Definition 4. *Let Λ be a lattice in $\mathbb{Z}^m$. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any parameter $\sigma \in \mathbb{R}_{>0}$, define*

$$\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \frac{\|\mathbf{x} - \mathbf{c}\|^2}{\sigma^2}) \text{ and } \rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x}).$$

The discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ is

$$\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}, \quad \text{for } \forall \mathbf{y} \in \Lambda.$$

If the center $\mathbf{c} = \mathbf{0}$, we conveniently use ρ_σ and $\mathcal{D}_{\Lambda,\sigma}$.

Micciancio and Regev [19] proved the following lemma that the norm of vectors sampled from the discrete Gaussian distribution is small with high probability. Gentry et al. [15] applied it to introduce lattice trapdoors.

Lemma 2. *For any m-dimensional integer lattice Λ, any vector $\mathbf{c} \in \mathbb{Z}^m$, any real $\epsilon \in (0, 1)$ and $\sigma \geq \omega(\sqrt{\log m})$,*

$$\Pr\left[\mathbf{x} \sim \mathcal{D}_{\Lambda,\sigma,\mathbf{c}} : \|\mathbf{x} - \mathbf{c}\| > \sqrt{m} \cdot \sigma\right] \leq \frac{1 + \epsilon}{1 - \epsilon} \cdot 2^{-m}.$$

Sampling Algorithms In [1, 4], there have been two algorithms proposed to sample short vectors from lattices. We will employ these two algorithms, SampleLeft and SampleRight, for our construction and security analysis, respectively, in a black-box manner. We demonstrate basic information of these two algorithms below.

$\mathsf{SampleLeft}(\mathbf{A}, \mathbf{M}, \mathbf{T_A}, \mathbf{u}, \sigma)$:

Input: a rank n matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$; a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times m_1}$; a short trapdoor basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$; a vector $\mathbf{u} \in \mathbb{Z}_q^n$; and a Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m + m_1)})$.

Output: a vector $\mathbf{e} \in \mathbb{Z}^{m+m_1}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}_1),\sigma}$, where $\mathbf{F}_1 = (\mathbf{A} \,|\, \mathbf{M})$. In particular, $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F}_1)$.

$\mathsf{SampleRight}(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T_B}, \mathbf{u}, \sigma)$:

Input: a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$; a rank n matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$; a uniform random matrix $\mathbf{R} \in \mathbb{Z}^{m \times m}$, let $s_{\mathbf{R}} := \|\mathbf{R}\|_R = \sup_{\|\mathbf{x}\|=1} \|\mathbf{R}\mathbf{x}\|$; a short trapdoor basis $\mathbf{T_B}$ of $\Lambda_q^{\perp}(\mathbf{B})$; a vector $\mathbf{u} \in \mathbb{Z}_q^n$; and a Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T_B}}\| \cdot s_{\mathbf{R}} \cdot \omega(\sqrt{\log m})$.

Output: a vector $\mathbf{e} \in \mathbb{Z}^{2m}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}_2),\sigma}$, where $\mathbf{F}_2 = (\mathbf{A} \,|\, \mathbf{AR} + \mathbf{B})$. In particular, $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F}_2)$.

It is known that the distributions of the outputs given by the two algorithms are statistically indistinguishable under the appropriate parameters when $m_1 = m$. See [1, 4] for further details of these algorithms.

Random Matrices in $\{-1, 1\}^{m \times k}$ For security analysis of our construction, we will use the following three lemmas about properties on random matrices in $\{-1, 1\}^{m \times k}$. The first lemma is a generalized version of the leftover hash lemma due to Dodis et al. [13].

Lemma 3. ([2, Lemma 13]) *Suppose that $m > (n + 1) \log q + \omega(\log n)$ and $q > 2$ is a prime. Let $\mathbf{A}, \mathbf{B}$ be matrices uniformly chosen from $\mathbb{Z}_q^{n \times m}$, and $\mathbb{Z}_q^{n \times k}$ respectively. Let $\mathbf{R}$ be an $m \times k$ matrix uniformly chosen from $\{-1, 1\}^{m \times k} \mod q$, where $k = k(n)$ is polynomial in n. Then for any vector $\mathbf{y} \in \mathbb{Z}_q^m$, the distribution of $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^{\top}\mathbf{y})$ is statistically close to the distribution of $(\mathbf{A}, \mathbf{B}, \mathbf{R}^{\top}\mathbf{y})$.*

The following lemmas bound the norm of a random matrix in $\{-1, 1\}^{m \times m}$, and the product of such a matrix with a vector, respectively.

Lemma 4. ([2, Lemma 15]) *Let $\mathbf{R}$ be a $m \times m$ matrix chosen at random from $\{-1, 1\}^{m \times m}$. Then there is a universal constant C such that*

$$\Pr\left[s_{\mathbf{R}} = \|\mathbf{R}\|_R > C\sqrt{m}\right] \leq e^{-2m}.$$

Lemma 5. ([2, Lemma 16]) *Let $\mathbf{R}$ be a $m \times m$ matrix chosen at random from $\{-1, 1\}^{m \times m}$. For any vector $\mathbf{y} \in \mathbf{R}^m$ we have*

$$\Pr\left[\|\mathbf{R}\mathbf{y}\| > \|\mathbf{y}\|\sqrt{m} \cdot \omega(\log m)\right] < \mathrm{negl}(m).$$

The LWE Hardness Assumption The learning with errors (LWE) problem, defined as follows, was first defined by Regev [21]. The security of our RIBE scheme is based on the hardness of this problem.

Definition 5 (LWE). *For a positive integer n, a prime $q = q(n)$ and a distribution χ over $\mathbb{Z}_q$, an $(\mathbb{Z}_q, n, \chi)$-LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being, either a noisy pseudo-random sampler $\mathcal{O}_{\mathbf{s}}$ under some constant random secret key $s \in \mathbb{Z}_q^n$, or a truly random sampler $\mathcal{O}_{\$}$. The behaviors of the two samplers are as follows:*

$\mathcal{O}_{\mathbf{s}}$: outputs samples of the form $(\mathbf{u}_i, v_i) = (\mathbf{u}_i, \mathbf{u}_i^{\top}\mathbf{s} + x_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniformly distributed vector which keeps persistent across invocations, $x_i \in \mathbb{Z}_q$ is a fresh sample form χ, and $\mathbf{u}_i$ is uniformly sampled from $\mathbb{Z}_q^n$.

$\mathcal{O}_{\$}$: outputs samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$ uniformly at random.

We state that an algorithm $\mathcal{B}$ decides the $(\mathbb{Z}_q, n, \chi)$-LWE problem if

$$\mathsf{Adv}_{\mathcal{B}}^{(\mathbb{Z}_q,n,\chi)\text{-LWE}} = \left|\Pr\left[\mathcal{B}^{\mathcal{O}_{\mathbf{s}}} = 1\right] - \Pr\left[\mathcal{B}^{\mathcal{O}_{\$}} = 1\right]\right|$$

is non-negligible for a random $\mathbf{s} \in \mathbb{Z}_q^n$.

Note that Regev [21] showed that for certain Gaussian error distribution χ, denoted by $\overline{\Psi}_\alpha$, the LWE problem is as hard as several standard worst-case lattice problems using a quantum reduction.

Definition 6. *For an $\alpha \in (0,1)$ and a prime q, let $\overline{\Psi}_\alpha$ denote the distribution over $\mathbb{Z}_q$ of the random variable $\lfloor qX \rceil \mod q$, where X is a normal distributed random variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, and $\lfloor x \rceil$ denotes the closest integer to x.*

Throughout the paper, $x \xleftarrow{\overline{\Psi}_\alpha} \mathbb{Z}_q$ (resp., $\mathbf{x} \xleftarrow{\overline{\Psi}_\alpha^m} \mathbb{Z}_q^m$) denotes that x (resp., $\mathbf{x}$) is selected from $\mathbb{Z}_q$ (resp., $\mathbb{Z}_q^m$) according to the distribution $\overline{\Psi}_\alpha$ (resp., $\overline{\Psi}_\alpha^m$).

Proposition 2 ([21]). *If there exists an efficient, possibly quantum, algorithm for deciding the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem for $\alpha \cdot q > 2\sqrt{n}$, then there exists a quantum with $q \cdot \mathsf{poly}(n)$-time algorithm for approximating the SIVP and GapSVP problems in the ℓ_2 norm, in the worst case, to within $\tilde{O}(n/\alpha)$ factors.*

As the best known algorithms for 2^k-approximations of gapSVP and SIVP run in time $2^{\widetilde{O}(n/k)}$ [14, 23], it derived from above that the LWE problem with noise ratio $\alpha = 2^{-n^\epsilon}$ is likely hard for some constant $\epsilon < 1$.

The following lemma, proposed in [2], on the distribution $\overline{\Psi}_\alpha$ will be used in parameter analysis in Section 3.3. The proof is implicitly deduced from Lemma 8.2 in [15].

Lemma 6. *Let $\mathbf{e} \in \mathbb{Z}^m$ and $\mathbf{y} \xleftarrow{\overline{\Psi}_\alpha^m} \mathbb{Z}_q^m$. Then the quantity $|\mathbf{e}^\top \mathbf{y}|$ regarded as an integer in $\{0\} \cup [q-1]$ satisfies*

$$|\mathbf{e}^\top \mathbf{y}| \leq \|\mathbf{e}\| q\alpha\omega(\sqrt{\log m}) + \|\mathbf{e}\|\sqrt{m}/2$$

with all but negligible probability in m.

3 RIBE from Lattices via Subset Difference Method

In this section, we provide our RIBE scheme from lattices via SD. Then, we look into the proper parameter sizes for our construction. The security analysis of our scheme will be given in Appendix A.

3.1 Our Construction

We begin by demonstrating some requirement about an identity and a time in our scheme: We treat an identity id and a time t as a sequence of length ℓ in $\{-1,1\}^\ell$, with possible redundancy, for some $\ell \in \mathbb{Z}$. The requirement is that for any tuple $(\mathsf{id}, \mathsf{t}) \in \mathcal{I} \times \mathcal{T}$, as two $\{-1,1\}$-vectors, are $\mathbb{Z}_q$-linearly independent. One possible technique to realize it is that if one sequence ends with 1, we put it in $\mathcal{T}$, otherwise in $\mathcal{I}$. Then for each sequence in $\mathcal{I} \cup \mathcal{T}$, we add extra 1 to the end. Thus, any $\mathsf{id} \in \mathcal{I}$ ends with 01, while any $\mathsf{t} \in \mathcal{T}$ ends with 11. As a result, the two vectors id, t are $\mathbb{Z}_q$-linearly independent.

Now, we describe our construction.

Setup$(1^\lambda, 1^N)$: On input a security parameter λ and the maximal number of users N, set the parameters $q, n, m, \sigma, \alpha, \ell$ as specified in Section 3.3 below. Then, it performs as follows:

1. Use the algorithm $\mathsf{TrapGen}(q,n)$ to generate a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a short basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$.
2. Select $2\ell+2$ uniformly random matrices $\mathbf{B}_1, \cdots, \mathbf{B}_\ell, \mathbf{C}; \mathbf{D}_1, \cdots, \mathbf{D}_\ell, \mathbf{G} \in \mathbb{Z}_q^{n \times m}$ and select a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$.
3. Let $\mathsf{UL}, \mathsf{RL}, \mathsf{FL}$ be three initially empty sets, which will be used to record the user list $\{(\mathsf{id}, \nu)\}$, revoked list $\{(\mathsf{id}, \mathsf{t})\}$, function list $\{(\mathsf{GL}_{d_j}^i, \mathsf{F}_{d_j}^i)\}$ respectively, where the domain of the parameters $\nu, \mathsf{GL}_{d_j}^i, \mathsf{F}_{d_j}^i$ will be specified in later steps.

4. Obtain a binary tree BT by running $\mathsf{SD.Setup}(N)$. Let $\mathcal{S}$ be the collection of all subsets $S_{i,j}$ of BT and $\mathsf{GL}^i_{d_j}$ be the subset of $\mathcal{S}$ consisting of $S_{i,j}$ such that the depth of node ν_j is d_j. Save $(\mathsf{GL}^i_{d_j}, \perp)$ to FL. Set the state to $\mathsf{st} = (\mathsf{BT}, \mathsf{UL})$.
5. Output the public parameters PP and master key MK given by

$$\mathsf{PP} = (\mathbf{A}, \{\mathbf{B}_1, \cdots, \mathbf{B}_\ell, \mathbf{C}\}, \{\mathbf{D}_1, \cdots, \mathbf{D}_\ell, \mathbf{G}\}, \mathbf{u}),$$
$$\mathsf{MK} = (\mathbf{T}_{\mathbf{A}}, \mathsf{FL}).$$

PriKeyGen$(\mathsf{PP}, \mathsf{MK}, \mathsf{id}, \mathsf{st})$: On input the public parameters PP, the master key MK, an identity $\mathsf{id} = (b_1, \cdots, b_\ell) \in \{-1, 1\}^\ell$, and the state st, it works as follows:

1. Randomly choose an unassigned leaf node ν in BT and assign it to the identity id. Save (id, ν) to UL. Run $\mathsf{SD.Assign}(\mathsf{BT}, \nu)$ to obtain $PV_\nu = \{S_{i,j}\}$.
2. Let $\mathbf{A}_{\mathsf{id}} = \mathbf{C} + \sum_{i=1}^{\ell} b_i \mathbf{B}_i \in \mathbb{Z}_q^{n \times m}$.
3. For each $S_{i,j} \in PV_\nu$, perform the following:
   - Retrieve the record $(\mathsf{GL}^i_{d_j}, *)$ from FL. If the second coordinate $*$ is not $\perp$, go to next step. Else, randomly select $\mathsf{F}^i_{d_j}(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $\mathsf{F}^i_{d_j}$ is a polynomial of degree 1 and $\mathsf{F}^i_{d_j}(0) = \mathbf{u}$. Update $(\mathsf{GL}^i_{d_j}, \mathsf{F}^i_{d_j})$ to FL.
   - Sample $\mathbf{e}^{\mathsf{id}}_{i,j}$ as

$$\mathbf{e}^{\mathsf{id}}_{i,j} \leftarrow \mathsf{SampleLeft}\left(\mathbf{A}, \mathbf{A}_{\mathsf{id}}, \mathbf{T}_{\mathbf{A}}, \mathsf{F}^i_{d_j}(j), \sigma\right).$$

   Note that for two sets $S_{i,j}, S_{i,j'} \in \mathsf{GL}^i_{d_j}$ with $j \neq j'$, we have $d_j = d_{j'}$. For given i and d_j, let $\mathsf{F} = \mathsf{F}^i_{d_j}$. We can compute fractional Lagrangian coefficients

$$L_j = j' \cdot (j' - j)^{-1} \quad \text{and} \quad L_{j'} = j \cdot (j - j')^{-1}$$

   such that $\mathbf{u} = \mathsf{F}(0) = L_j \mathsf{F}(j) + L_{j'} \mathsf{F}(j') \mod q$. That is, as a fraction of integers, L_j can also be evaluated in $\mathbb{Z}_q$, then L_j can be regarded as an integer in $\{0\} \cup [q-1]$.
4. Output the updated state st and the private key $\mathsf{sk}_{\mathsf{id}}$ for id as

$$\left(PV_\nu, \{\mathbf{e}^{\mathsf{id}}_{i,j}\}_{S_{i,j} \in PV_\nu}\right).$$

UpdateKey$(\mathsf{PP}, \mathsf{MK}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$: On input the public parameters PP, the master key MK, a time $\mathsf{t} = (t_1, \cdots, t_\ell) \in \{-1, 1\}^\ell$, the revoked list RL, and the state st, it works as follows:

1. Take R as a set consisting of such id's that for some $\mathsf{t}' \leq \mathsf{t}$, $(\mathsf{id}, \mathsf{t}') \in \mathsf{RL}$. By using UL, define RI as the set of index of leaf nodes corresponding the id's in R. That is, for any $\mathsf{id} \in R$, if $(\mathsf{id}, \nu) \in \mathsf{UL}$, then put ν in RI.
2. Run $\mathsf{SD.Cover}(\mathsf{BT}, RI)$ to obtain $CV_{\mathrm{RI}} = \{S_{i,j}\}$.
3. Let $\mathbf{A}_{\mathsf{t}} = \mathbf{G} + \sum_{i=1}^{\ell} t_i \mathbf{D}_i \in \mathbb{Z}_q^{n \times m}$.
4. For each $S_{i,j} \in CV_{\mathrm{RI}}$, perform the following:
   - Retrieve the record $(\mathsf{GL}^i_{d_j}, *)$ from FL. If the second coordinate $*$ is not $\perp$, go to next step. Else, randomly select $\mathsf{F}^i_{d_j}(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $\mathsf{F}^i_{d_j}$ is a polynomial of degree 1 and $\mathsf{F}^i_{d_j}(0) = \mathbf{u}$. Update $(\mathsf{GL}^i_{d_j}, \mathsf{F}^i_{d_j})$ to FL.
   - Sample $\mathbf{e}^{\mathsf{t}}_{i,j}$ as

$$\mathbf{e}^{\mathsf{t}}_{i,j} \leftarrow \mathsf{SampleLeft}\left(\mathbf{A}, \mathbf{A}_{\mathsf{t}}, \mathbf{T}_{\mathbf{A}}, \mathsf{F}^i_{d_j}(j), \sigma\right).$$

5. Output the update key uk_{t} at time t as

$$\left(CV_{\mathrm{RI}}, \{\mathbf{e}^{\mathsf{t}}_{i,j}\}_{S_{i,j} \in CV_{\mathrm{RI}}}\right).$$

Encrypt$(\mathsf{PP}, \mathsf{id}, \mathsf{t}, b)$: On input the public parameters PP, an identity id, a current time t, a message $b \in \{0, 1\}$, it performs as follows:

1. Let $D = (N-1)!$.
2. Choose a uniformly random vector $\mathbf{s} \in \mathbb{Z}_q^n$.

3. Choose 2ℓ uniformly random matrixes $\mathbf{R}_i \in \{-1,1\}^{m\times m}$ for $i = 1, \cdots, 2\ell$ and define $\mathbf{R}_{\mathsf{id}} = \sum_{i=1}^{\ell} b_i \mathbf{R}_i$, $\mathbf{R}_{\mathsf{t}} = \sum_{i=1}^{\ell} t_i \mathbf{R}_{\ell+i} \in \{-\ell, \cdots, \ell\}^{m\times m}$.
4. Choose a noise value $x \xleftarrow{\overline{\Psi}_\alpha} \mathbb{Z}_q$ and a noise vector $\mathbf{y} \xleftarrow{\overline{\Psi}_\alpha^m} \mathbb{Z}_q^m$. Set

$$\mathbf{x}_{\mathsf{id}} = \begin{pmatrix} \mathbf{y} \\ \mathbf{R}_{\mathsf{id}}^\top \mathbf{y} \end{pmatrix} \quad \text{and} \quad \mathbf{x}_{\mathsf{t}} = \begin{pmatrix} \mathbf{y} \\ \mathbf{R}_{\mathsf{t}}^\top \mathbf{y} \end{pmatrix}.$$

5. Compute

$$\begin{aligned} c_0 &\leftarrow \mathbf{u}^\top \mathbf{s} + Dx + b\left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q, \\ \mathbf{c}_1 &\leftarrow (\mathbf{A} \,|\, \mathbf{A}_{\mathsf{id}})^\top \mathbf{s} + D\mathbf{x}_{\mathsf{id}} \in \mathbb{Z}_q^{2m}, \\ \mathbf{c}_2 &\leftarrow (\mathbf{A} \,|\, \mathbf{A}_{\mathsf{t}})^\top \mathbf{s} + D\mathbf{x}_{\mathsf{t}} \in \mathbb{Z}_q^{2m}. \end{aligned}$$

and output the ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (c_0, \mathbf{c}_1, \mathbf{c}_2)$.

DecKeyGen$(\mathsf{PP}, \mathsf{sk}_{\mathsf{id}}, \mathsf{uk}_{\mathsf{t}})$: On input the public parameters PP, a private key $\mathsf{sk}_{\mathsf{id}}$ and an update key uk_{t} given by

$$\begin{aligned} \mathsf{sk}_{\mathsf{id}} &= \left(PV_\nu, \{\mathbf{e}_{i,j}^{\mathsf{id}}\}_{S_{i,j} \in PV_\nu}\right) \text{ and} \\ \mathsf{uk}_{\mathsf{t}} &= \left(CV_{\mathrm{RI}}, \{\mathbf{e}_{i,j}^{\mathsf{t}}\}_{S_{i,j} \in CV_{\mathrm{RI}}}\right), \end{aligned}$$

it performs as follows:

1. If $\mathsf{id} \notin R$, run $\mathsf{SD.Match}(CV_{\mathrm{RI}}, PV_\nu)$ to obtain a set pair $(S_{i,j}, S_{i',j'})$ such that $S_{i,j} \in CV_{\mathrm{RI}}$, $S_{i',j'} \in PV_\nu$, and $(i = i') \wedge (d_j = d_{j'}) \wedge (j \neq j')$. Else, output $\perp$.
2. Retrieve $\mathbf{e}_{i',j'}^{\mathsf{id}}$ from $\mathsf{sk}_{\mathsf{id}}$ and $\mathbf{e}_{i,j}^{\mathsf{t}}$ from uk_{t}.
3. Compute the fractional Lagrangian coefficients L_j and $L_{j'}$ and evaluate them in $\mathbb{Z}_q$, that is, they can be interpreted as elements in $\mathbb{Z}_q$, i.e. integers in $\{0\} \cup [q-1]$.
4. Output the decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = \left(L_{j'} \mathbf{e}_{i',j'}^{\mathsf{id}}, L_j \mathbf{e}_{i,j}^{\mathsf{t}}\right)$.

Decrypt$(\mathsf{PP}, \mathsf{ct}, \mathsf{dk}_{\mathsf{id},\mathsf{t}})$: On input the public parameters PP, a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$, and a decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$, it performs as follows:

1. Evaluate $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$ in $\mathbb{Z}_q^{2m}$ as $(\mathbf{d}_1, \mathbf{d}_2)$, where $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{Z}_q^m$.
2. Compute $w \leftarrow c_0 - \mathbf{d}_1^\top \mathbf{c}_1 - \mathbf{d}_2^\top \mathbf{c}_2 \mod q$.
3. Compare w and $\left\lfloor \frac{q}{2} \right\rfloor$ regarding them as integers in $\mathbb{Z}$. If $\left|w - \left\lfloor \frac{q}{2} \right\rfloor\right| < \left\lfloor \frac{q}{4} \right\rfloor$ output 1. Else, output 0.

Revoke$(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$: On input an identity id, a revocation time t, the revocation list RL, and the state $\mathsf{st} = (\mathsf{BT}, \mathsf{UL})$: If $(\mathsf{id}, *) \notin \mathsf{UL}$, then output $\perp$ as the private key for id was not generated. Else, add $(\mathsf{id}, \mathsf{t})$ to RL. Then output the updated revocation list RL.

3.2 Correctness

Now, we demonstrate the correctness of our construction. Let $\mathsf{F} = \mathsf{F}_{d_j}^i$. If the ciphertext is generated by operating as described above, we can compute the error term in the decryption algorithm from the following relation:

$$\begin{aligned} w &= c_0 - L_{j'} \left(\mathbf{e}_{i',j'}^{\mathsf{id}}\right)^\top \mathbf{c}_1 - L_j \left(\mathbf{e}_{i,j}^{\mathsf{t}}\right)^\top \mathbf{c}_2 \\ &= \mathbf{u}^\top \mathbf{s} - \left[L_{j'} \mathsf{F}(j') + L_j \mathsf{F}(j)\right]^\top \mathbf{s} + Dx + b\left\lfloor \frac{q}{2} \right\rfloor \\ &\quad - DL_{j'} \left(\mathbf{e}_{i',j'}^{\mathsf{id}}\right)^\top \mathbf{x}_{\mathsf{id}} - DL_j \left(\mathbf{e}_{i,j}^{\mathsf{t}}\right)^\top \mathbf{x}_{\mathsf{t}} \\ &= b\left\lfloor \frac{q}{2} \right\rfloor + \underbrace{Dx - DL_{j'} \left(\mathbf{e}_{i',j'}^{\mathsf{id}}\right)^\top \mathbf{x}_{\mathsf{id}} - DL_j \left(\mathbf{e}_{i,j}^{\mathsf{t}}\right)^\top \mathbf{x}_{\mathsf{t}}}_{\text{error term}}, \end{aligned}$$

where all the equalities are in $\mathbb{Z}_q$.

Hence, for the correctness of our construction, it suffices to set the parameters such that

$$\begin{aligned}&|Dx - DL_{j'}\left(\mathbf{e}^{\mathsf{id}}_{i',j'}\right)^\top \mathbf{x}_{\mathsf{id}} - DL_j\left(\mathbf{e}^{\mathsf{t}}_{i,j}\right)^\top \mathbf{x}_{\mathsf{t}}|\\ &\leq (N!)\left(|x| + 2\left|\left(\mathbf{e}^{\mathsf{id}}_{i',j'}\right)^\top \mathbf{x}_{\mathsf{id}}\right| + 2\left|\left(\mathbf{e}^{\mathsf{t}}_{i,j}\right)^\top \mathbf{x}_{\mathsf{t}}\right|\right) \leq q/5, \end{aligned} \tag{1}$$

except with negligible probability. For the first inequality, as j and j' have the same depth, we know $|j - j'| \leq N - 1$, and hence $j - j'$ divides D. Moreover, j, j' lie in the interval $[2N - 1]$, so $DL_j, DL_{j'}$ are integers not exceeding $(N-1)! \times (2N-1) < 2\,(N!)$. Therefore, if we set the parameters such that q is sufficiently larger than $N!$, then our construction satisfies the correctness.

We will give details of parameter sizes in Section 3.3.

3.3 Parameters

To bound the error term in the decryption algorithm of our construction, we first introduce the following lemma. The proof of this lemma is straightforward from Lemma 4 and Lemma 6, and a similar case has been analyzed in Lemma 24 in [2]. We omit the details of the proof.

Lemma 7. *The norm of the error term in Section 3.2 is bounded by*

$$4q\sigma\ell m\alpha \cdot \omega(\sqrt{\log m}) \cdot (N!) + 4O(\sigma m^{1.5}) \cdot (N!)$$

except for a negligible probability.

Now we set the parameters to guarantee that the decryption is correct and the security reduction is meaningful. The parameters are set under the following requirements:

1. For the lattice trapdoor generation algorithm in Proposition 1, the parameters should satisfy $m \geq 2n \log q$. Under this selection of m, the output basis of $\mathsf{TrapGen}$ has Gram-Schmidt norm at most $O(\sqrt{m})$. The private key $\mathsf{sk}_{\mathsf{id}}$ and the update key uk_{t} are generated from the algorithm $\mathsf{SampleLeft}$ in real scheme and $\mathsf{SampleRight}$ in simulated games, respectively. The two algorithms are presented in Section 2.3. By Lemma 2, if we set Gaussian parameter $\sigma \geq \ell m \log m$, then the vectors inside the two keys have length at most $\sigma\sqrt{2m} \leq 2\ell m^{1.5} \log m$ with high probability.
2. The noise distribution is set as $\chi = \overline{\Psi}^m_\alpha$, where $\alpha \geq 2\sqrt{m}/q$ in order to apply Regev's reduction (see Proposition 2). Any vector $\mathbf{y}$ sampled from this distribution has length satisfying $O(\alpha q\sqrt{m}) \leq 2m$ with high probability.
3. The norm of the error term is bounded by

$$\begin{aligned}&(N!)\left(|x| + 2\left|\left(\mathbf{e}^{\mathsf{id}}_{i',j'}\right)^\top \mathbf{x}_{\mathsf{id}}\right| + 2\left|\left(\mathbf{e}^{\mathsf{t}}_{i,j}\right)^\top \mathbf{x}_{\mathsf{t}}\right|\right)\\ &\leq 4q\sigma\ell m\alpha \cdot \omega(\sqrt{\log m}) \cdot (N!) + 4O(\sigma m^{1.5}) \cdot (N!)\\ &\leq 16\ell^2(N!) \cdot m^{2.5} \cdot \log m \cdot \omega(\sqrt{\log m})\\ &\qquad\qquad +4(N!) \cdot O(\ell m^{2.5} \log m)\\ &\leq 16\ell^2(N!) \cdot m^{2.5}(\log m)^{1.5}\\ &\leq 2^{N \log N}\ell^2 \cdot m^{2.5} \cdot (\log m)^{1.5}\end{aligned}$$

   where we used the fact that $N! \leq 2^{N \log N}$. The modulus q satisfying $q \geq 2^{N \log N}\ell^2 \cdot m^{2.5} \cdot (\log m)^{1.5}$ ensures the correctness of our construction.
4. The modulus q should satisfy $q \geq 2(Q^{\mathsf{id}} + |\mathcal{T}|)$, where Q^{id} is the number of identity queries from the adversary, $|\mathcal{T}|$ is the size of time space. This requirement ensures that our reduction applies. The details are given in Appendix A.
5. The identity space $\mathcal{I}$ is sufficient for N users, so $2^\ell \geq N$.

To satisfy these requirements, given a constant $\epsilon \in (0, 1)$, we set the parameters as follows, taking n to be the security parameter:

$$n = (N \log N)^{1/\epsilon}, \quad \ell = \log N, \tag{2}$$

$$m = 2n^{1+\delta}, \quad q \geq \max\left\{2\left(Q^{\mathsf{id}} + |\mathcal{T}|\right), 2^{N \log N}\ell^2 \cdot n^5\right\}, \tag{3}$$

$$\sigma = \ell n^2 \log n, \quad \alpha = \left[2^{N \log N}\ell^2 \cdot n^4\right]^{-1}, \tag{4}$$

where q is the nearest larger prime, and δ is selected such that $1 > \delta > \epsilon$.

Note that from Regev's reduction (see Proposition 2), the security of our scheme is finally based on the hardness of $2^{O(n^\epsilon)}$-approximating gapSVP or SIVP on n-dimension lattices using algorithms that run in time $q \cdot \mathsf{poly}(n) = 2^{O(n^\epsilon)}$. The same assumption has been used to construct fuzzy IBE by Agrawal et al. in [4]. The security holds for $\epsilon < 1/2$.

4 Conclusion and Open problems

We have provided the first adaptive-ID secure revocable identity-based encryption scheme from lattices in the standard model under the LWE assumption. To realize identity revocation, our scheme employs the subset difference method.

However, unfortunately, our construction can not achieve to thwart the decryption key exposure attack, proposed by Seo and Emura [24], which considers not only exposure of a long-term private key and a short-term update key, but also exposure of a short-term decryption key. To directly apply Seo and Emura's approach which randomizes a short-term decryption key, we need a proper one-way function to randomize the short-term decryption key in our scheme, but we could not find such a proper one-way function and we leave it as an open problem. In our scheme, we just put $\mathsf{sk}_{\mathsf{id}}$ and uk_{t} together to generate the decryption key. If we have a proper one-way function mapping the pair of vectors to another pair which also satisfy the dual-Regev encryption scheme, and from the new pair it is hard to regain $\mathsf{sk}_{\mathsf{id}}$ and uk_{t}, then our scheme can be modified to achieve the security against the decryption key exposure attack.

It would be also interesting to construct revocable identity-based signatures and attribute-based encryption schemes from lattices.

Acknowledgments. The authors are particularly grateful to Khoa Nguyen and Hyung Tae Lee for the helpful discussions. Meanwhile, the authors would like to thank San Ling, Huaxiong Wang, Chaoping Xing for their support all along, and the anonymous referees for the valuable comments. The first author is supported by NTU Research Scholarship at Nanyang Technological University, while the second author is supported in part by the Singapore Ministry of Education under Research Grant MOE2013-T2-1-041.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient Lattice (H)IBE in the Standard Model. In H. Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.
2. S. Agrawal, D. Boneh, and X. Boyen. Efficient Lattice (H)IBE in the Standard Model. Full version of [1]. Avaiable at `http://crypto.stanford.edu/~dabo/pubs/papers/latticebb.pdf`, 2010.
3. S. Agrawal, D. Boneh, and X. Boyen. Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In T. Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 98–115. Springer, 2010.
4. S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee. Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *Public Key Cryptography - PKC 2012*, volume 7293 of *Lecture Notes in Computer Science*, pages 280–297. Springer, 2012.
5. M. Ajtai. Generating Hard Instances of the Short Basis Problem. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.
6. J. Alwen and C. Peikert. Generating Shorter Bases for Hard Random Lattices. In S. Albers and J. Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009*, volume 3 of *LIPIcs*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.
7. A. Boldyreva, V. Goyal, and V. Kumar. Identity-based Encryption with Efficient Revocation. In P. Ning, P. F. Syverson, and S. Jha, editors, *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008*, pages 417–426. ACM, 2008.
8. D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

9. D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
10. D. Boneh, C. Gentry, and M. Hamburg. Space-Efficient Identity Based Encryption Without Pairings. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 647–657. IEEE Computer Society, 2007.
11. J. Chen, H. W. Lim, S. Ling, H. Wang, and K. Nguyen. Revocable Identity-Based Encryption from Lattices. In W. Susilo, Y. Mu, and J. Seberry, editors, *Information Security and Privacy - 17th Australasian Conference, ACISP 2012*, volume 7372 of *Lecture Notes in Computer Science*, pages 390–403. Springer, 2012.
12. C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In B. Honary, editor, *Cryptography and Coding, 8th IMA International Conference*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.
13. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.*, 38(1):97–139, 2008.
14. N. Gama and P. Q. Nguyen. Finding Short Lattice Vectors within Mordell's Inequality. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008*, pages 207–216. ACM, 2008.
15. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008*, pages 197–206. ACM, 2008.
16. K. Lee, D. H. Lee, and J. H. Park. Efficient Revocable Identity-Based Encryption via Subset Difference Methods. IACR Cryptology ePrint Archive 2014:132. Avaiable at `http://eprint.iacr.org/2014/132`, 2014.
17. B. Libert and D. Vergnaud. Adaptive-ID Secure Revocable Identity-Based Encryption. In M. Fischlin, editor, *Topics in Cryptology - CT-RSA 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2009.
18. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012.
19. D. Micciancio and O. Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM J. Comput.*, 37(1):267–302, 2007.
20. D. Naor, M. Naor, and J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.
21. O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In H. N. Gabow and R. Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, 2005*, pages 84–93. ACM, 2005.
22. A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.
23. C. Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.
24. J. H. Seo and K. Emura. Revocable Identity-Based Cryptosystem Revisited: Security Models and Constructions. *Information Forensics and Security, IEEE Transactions on*, 9(7):1193–1205, July 2014.
25. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1985.
26. V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2008.
27. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.
28. B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.

A Security Analysis

Our RIBE scheme defined in Section 3.1 with parameters $(q, n, m, \sigma, \alpha, \ell)$ as in (2)-(4), is indistinguishable from random under adaptive identity-time attacks provided that the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE assumption holds. In particular, we have the following theorem.

Theorem 1. *Under the condition that Q^{id} the number of private key queries and $|\mathcal{T}|$ the size of time space satisfy $2(Q^{id} + |\mathcal{T}|) \leq q$, and both of them are polynomial size of n, if there exists a PPT adversary $\mathcal{A}$ that wins INDr-RID-CPA game with advantage ϵ, then there exists a PPT algorithm $\mathcal{B}$ that solves the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem in about the same time as $\mathcal{A}$ and with advantage $\epsilon' \geq \epsilon/(8q^2 Q^{id} \cdot |\mathcal{T}|)$.*

A.1 Abort-resistant hash functions

Similar as [1], the proof of Theorem 1 requires *abort-resistant hash functions* to implement abort events in indistinguishable games. We adaptively modify the definition and propositions of abort-resistant hash functions from [2].

Definition 7. *Let $\mathcal{H} := \{H : X \to Y\}$ be a family of hash functions from X to Y where $0 \in Y$. For a set of Q inputs $\overline{x} = (x_1, \cdots, x_Q) \in X^Q$, define the non-abort probability in terms of $\mathfrak{I}$-place of $\overline{x}$ as*

$$\alpha_{\mathfrak{I}}(\overline{x}) := \Pr\left[H \in \mathcal{H} : E_{\mathfrak{I}}(H)\right],$$

where $E_{\mathfrak{I}}(H)$ is the event that given $\mathfrak{I} \subseteq \{1, \cdots, Q\}$,

$$(H(x_i) = 0, \forall i \in \mathfrak{I}) \wedge (H(x_k) \neq 0, \forall k \in [Q] \setminus \mathfrak{I}), \tag{5}$$

and the probability is over the random choice of H in $\mathcal{H}$. We call that $\mathcal{H}$ is $(Q, \alpha_{\mathfrak{I},\min}, \alpha_{\mathfrak{I},\max})$ abort-resistant *in terms of $\mathfrak{I}$-place if for all $\overline{x} \in X^Q$ with the condition $\{x_i : i \in \mathfrak{I}\} \cap \{x_k : k \in [Q] \setminus \mathfrak{I}\} = \emptyset$, we have $\alpha_{\mathfrak{I}}(\overline{x}) \in [\alpha_{\mathfrak{I},\min}, \alpha_{\mathfrak{I},\max}]$.*

We will employ the following abort-resistant hash family utilized in [28, 1]. For a prime q let $(\mathbb{Z}_q^\ell)^* := \mathbb{Z}_q^\ell \setminus \{0^\ell\}$ and define the family $\mathcal{H}_{\mathrm{Wat}} := \{H_h : (\mathbb{Z}_q^\ell)^* \to \mathbb{Z}_q\}_{h \in \mathbb{Z}_q^\ell}$ as

$$H_h(\mathbf{e}) := 1 + \sum_{i=1}^{\ell} h_i e_i \in \mathbb{Z}_q \tag{6}$$

where $\mathbf{e} = (e_1, \cdots, e_\ell) \in (\mathbb{Z}_q^\ell)^*$ and $h = (h_1, \cdots, h_\ell) \in \mathbb{Z}_q^\ell$.

In our proof, the inputs of these hash functions will be only in $\{-1, 1\}^\ell$, a smaller domain than the general domain where abort-resistance holds. Furthermore, in our reduction, we need the non-abort probability in terms of $\mathfrak{I}$-place with $|\mathfrak{I}| = 1$ or 2.

Lemma 8. *Let q be a prime, $0 < Q < q$ and $\mathfrak{I}$ are fixed subset of $[Q]$. Then the hash family $\mathcal{H}_{Wat}$ defined in (6) satisfies*

1. *if $\mathfrak{I} = \{i\}$, $\mathcal{H}_{Wat}$ is $\left(Q, \frac{1}{q}\left(1 - \frac{Q}{q}\right), \frac{1}{q}\right)$ abort-resistant in terms of $\{i\}$-place.*
2. *if $\mathfrak{I} = \{i, j\}$, and $\mathbf{e}_i, \mathbf{e}_j$ are $\mathbb{Z}_q$-linearly independent, $\mathcal{H}_{Wat}$ is $\left(Q, \frac{1}{q^2}\left(1 - \frac{Q}{q}\right), \frac{1}{q^2}\right)$ abort-resistant in terms of $\{i, j\}$-place.*

Proof. The first part for $|\mathfrak{I}| = 1$ has been proved in [1]. For $\mathfrak{I} = \{i, j\}$, we consider a set of inputs $\mathbf{e}_1, \cdots, \mathbf{e}_Q \in (\mathbb{Z}_q^\ell)^*$. For any $k \in [Q]$, let S_k be the set of functions in $\mathcal{H}_{\mathrm{Wat}}$ such that $H(\mathbf{e}_k) = 0$. As $\mathbf{e}_i, \mathbf{e}_j$ are $\mathbb{Z}_q$-linearly independent, $|S_i \cap S_j| = q^{\ell-2}$. Moreover, for $k \notin \{i, j\}$, as $\mathbf{e}_k \neq \mathbf{e}_i$ or $\mathbf{e}_j$ and all coordinates of $\mathbf{e}_k$ are -1 or 1, if $\mathbf{e}_k = s\mathbf{e}_i + t\mathbf{e}_j$ for $s, t \in \mathbb{Z}_q$, then one of s, t must be zero. This implies $\mathbf{e}_k = -\mathbf{e}_i$ or $-\mathbf{e}_j$, or $\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k$ are $\mathbb{Z}_q$-linearly independent. So $|S_i \cap S_j \cap S_k| \leq q^{\ell-3}$. The set of functions in $\mathcal{H}_{\mathrm{Wat}}$, such that $E_{\mathfrak{I}}(H)$ defined in (5) holds, is exactly $S := (S_i \cap S_j) \setminus \cup_{k \neq i,j} S_k$. By inclusion-exclusion principle, we have $|S| \geq q^{\ell-2} - Qq^{\ell-3}$. Therefore the non-abort probability in terms of (i, j)-place, which equals $|S|/q^\ell$, is at least $\left(1 - \frac{Q}{q}\right)/q^2$, at most $|S_i \cap S_j|/q^\ell \leq q^{\ell-2}/q^\ell = 1/q^2$.

A.2 Proof of Theorem 1

In this subsection we prove the security of our scheme against adaptive adversary. In Game i, let W_i denote the event that the adversary correctly guesses the challenge bit, i.e. $r' = r$ at the end of the guess phase. The adversary's advantage in Game i is denoted by $\left|\Pr[W_i] - \frac{1}{2}\right|$.

Game 0. This is the original INDr-RID-CPA game from Definition 1 between an attacker against our scheme and an INDr-RID-CPA challenger.

Game 1. In Game 1 we change the way that the challenger generates the matrices $\mathbf{B}_i, \mathbf{D}_i$, $i \in [\ell]$. The Game 1 challenger selects $\mathbf{R}_i^*$, $i \in [2\ell]$ at the setup phase and also chooses 2ℓ random scalars $h_i \in \mathbb{Z}_q$ for $i \in [2\ell]$. Next it generates $\mathbf{A}$, $\mathbf{C}$, $\mathbf{G}$ as in Game 0 and constructs the matrices $\mathbf{B}_i, \mathbf{D}_i$ for $i \in [\ell]$ as

$$\mathbf{B}_i \leftarrow \mathbf{A} \cdot \mathbf{R}_i^* + h_i \mathbf{C} \text{ and } \mathbf{D}_i \leftarrow \mathbf{A} \cdot \mathbf{R}_{\ell+i}^* + h_{l+i} \mathbf{G}. \tag{7}$$

The remainder of the game is unchanged. Note that the knowledge of the challenge identity id^* and time t^* is not needed for such setup.

We prove $\Pr[W_0]$ =$\Pr[W_1]$. The $\mathbf{R}_i^*$, $i \in [2\ell]$ are only used in the construction of the matrices $\mathbf{B}_i, \mathbf{D}_i$ and in the construction of the challenge ciphertext where $\mathbf{z}_{\mathsf{id}} \leftarrow (\mathbf{R}_{\mathsf{id}}^*)^\top \mathbf{y}$, $\mathbf{z}_{\mathsf{t}} \leftarrow (\mathbf{R}_{\mathsf{t}}^*)^\top \mathbf{y}$, and $\mathbf{R}_{\mathsf{id}^*}^* = \sum_{i=1}^{\ell} b_i^* \mathbf{R}_i^*$, $\mathbf{R}_{\mathsf{t}^*}^* = \sum_{i=1}^{\ell} t_i^* \mathbf{R}_{\ell+i}^*$. Let $\mathbf{R}^* = (\mathbf{R}_1^* | \cdots | \mathbf{R}_\ell^*) \in \mathbb{Z}_q^{m \times \ell m}$, then by Lemma 3, the distributions

$$\left(\mathbf{A}, \mathbf{A} \cdot \mathbf{R}^*, (\mathbf{R}^*)^\top \mathbf{y}\right) \qquad \text{and}$$
$$\left(\mathbf{A}, (\mathbf{B}_1' | \cdots | \mathbf{B}_\ell' | \mathbf{D}_1' | \cdots | \mathbf{D}_\ell'), (\mathbf{R}^*)^\top \mathbf{y}\right)$$

are statistically close, where $\mathbf{B}_i', \mathbf{D}_i'$ $i \in [\ell]$ are uniformly independent matrices in $\mathbb{Z}_q^{n \times m}$. Then it follows that with $\mathbf{z} = (\mathbf{z}_{\mathsf{id}} \| \mathbf{z}_{\mathsf{t}})$, the distributions

$$\left(\mathbf{A}, \mathbf{A}\mathbf{R}_1^*, \cdots, \mathbf{A}\mathbf{R}_\ell^*, \mathbf{A}\mathbf{R}_{\ell+1}^*, \cdots, \mathbf{A}\mathbf{R}_{2\ell}^*, \mathbf{z}\right) \text{ and}$$
$$\left(\mathbf{A}, \mathbf{B}_1', \cdots, \mathbf{B}_\ell', \mathbf{D}_1', \cdots, \mathbf{D}_\ell', \mathbf{z}\right)$$

are statistically close. Therefore in the adversary's view, the matrices $\mathbf{A}\mathbf{R}_i^*$ are statistically close to uniform and independent of $\mathbf{z}$. Hence the $\mathbf{B}_i, \mathbf{D}_i$ defined in (7) are close to uniform, thus

$$\Pr[W_0] = \Pr[W_1]. \tag{8}$$

Game 2. Game 2 is the same as Game 1 except we add an abort event that is hidden from the adversary's view. Let Q^{id} be the maximal number of queries regarding private key queries, and $|\mathcal{T}|$ the size of time space which are polynomial size in n. Let $q \geq 2\left(|\mathcal{T}| + Q^{\mathsf{id}}\right)$. Without loss of generality, since the time space $\mathcal{T}$ has size polynomial in λ, we assume adversary $\mathcal{A}$ queries the update keys for all time $\mathsf{t} \in \mathcal{T}$. And the challenger can return the same response to repeated queries from local records, so we may assume $(\mathsf{t}_1, \cdots, \mathsf{t}_{|\mathcal{T}|})$ is the time queries tuple, which is listed in an increasing order since the update key can not be queried in a decreasing order.

The challenger of Game 2 behaves as follows:

1. The setup phase is identical to Game 1 except that the challenger also selects a random hash function $\tilde{H} \in \mathcal{H}_{\mathrm{Wat}}$ and keeps it to itself.
2. The challenger responds to the update key queries and issues the challenge ciphertext exactly as in Game 1.
3. The Game 2 challenger randomly guesses $i^* \in [|\mathcal{T}|]$ such that $\mathcal{A}$'s i^*-th update key query is for time t^*. The challenger guesses it correctly with probability $1/|\mathcal{T}|$.
4. The Game 2 challenger guesses an adversary type from the two following types:
   i. Type-1 adversary: $\mathcal{A}$ does not query $\mathsf{sk}_{\mathsf{id}^*}$.
   ii. Type-2 adversary: $\mathcal{A}$ makes private key query for the challenge identity id^*, so id^* should be revoked before time t^*.

   The challenger guesses it correctly with probability $1/2$.
   If the guess is Type-2, the Game 2 challenger randomly guesses $j' \in [Q^{\mathsf{id}}]$ such that $\mathcal{A}$'s j'-th private key query is for id^*. The challenger guesses it correctly with probability $1/Q^{\mathsf{id}}$. Let $j^* = |\mathcal{T}| + j'$.
5. Set $Q = Q^{\mathsf{id}} + |\mathcal{T}|$. In the final guess phase, the adversary outputs a guess $r' \in \{0, 1\}$. The challenger does the following:
   a. Set $\mathbb{e} = (\mathbf{e}_1, \cdots, \mathbf{e}_Q) = (\mathsf{t}_1, \cdots, \mathsf{t}_{|\mathcal{T}|}, \mathsf{id}_1, \cdots, \mathsf{id}_{Q^{\mathsf{id}}})$, where $(\mathsf{t}_1, \cdots, \mathsf{t}_{|\mathcal{T}|})$ and $(\mathsf{id}_1, \cdots, \mathsf{id}_{Q^{\mathsf{id}}})$ are the queried time tuple and the identity tuple respectively. Set

$$\mathfrak{I} = \begin{cases} \{i^*\}, \text{ if } \mathcal{A} \text{ is guessed Type-1;} \\ \{i^*, j^*\}, \text{ if } \mathcal{A} \text{ is guessed Type-2.} \end{cases}$$

   b. **Abort check:** on input $\mathbb{e}$, the challenger checks if $\tilde{H}$ satisfies $E_{\mathfrak{I}}(\tilde{H})$ defined in (5). If not, it overwrites r' with a fresh random bit in $\{0, 1\}$ and we say the challenger aborted the game. Note that the adversary never sees $\tilde{H}$ and has no idea if the game ends with an abort.

c. **Artificial abort:** the challenger samples a bit $\Gamma \in \{0,1\}$ such that $\Pr[\Gamma = 1] = \gamma(\mathbb{e})$ where the analysis of such artificial abort and the definition of the function $\gamma(\cdot)$ are given in [28]. If $\Gamma = 1$, the challenger overwrites r' with a fresh random bit in $\{0,1\}$ and we say the challenger aborted the game due to an artificial abort. The reason for this step will be briefly explained later, while for the details one may refer to [28, 1].

For queries tuple $\mathbb{e}$, let $\epsilon(\mathbb{e})$ be the probability that an abort (either real or artificial) does not happen. Let $\epsilon^{(b)}_{\max}$ and $\epsilon^{(b)}_{\min}$ be scalars such that $\epsilon(\mathbb{e}) \in [\epsilon^{(b)}_{\min}, \epsilon^{(b)}_{\max}]$ for the case $\mathcal{A}$ is guessed Type-b, $b = 1, 2$.

Lemma 9. ([2, Lemma 28]) *For Type-b adversary, we have that*

$$\left| Pr[W_2] - \frac{1}{2} \right| \geq \epsilon^{(b)}_{\min} \left| Pr[W_1] - \frac{1}{2} \right| - \frac{1}{2}(\epsilon^{(b)}_{\max} - \epsilon^{(b)}_{\min}).$$

In our case, $\epsilon(\mathbb{e})$ is the probability that the Game 2 does not abort. Set

$$\Delta_b = \begin{cases} \frac{1}{2|\mathcal{T}|}, & \text{if } b = 1; \\ \frac{1}{2|\mathcal{T}| \cdot Q^{\mathsf{id}}}, & \text{if } b = 2, \end{cases}$$

i.e. the probability that the challenger guesses the type of adversary, $\mathfrak{I}$-place correctly.

If there is no artificial abort, i.e. $\gamma(\cdot)$ is identically zero, then $\epsilon^{(b)}_{\min} = \Delta_b \cdot \alpha_{\mathfrak{I}_b,\min}$, $\epsilon^{(b)}_{\max} = \Delta_b \cdot \alpha_{\mathfrak{I}_b,\max}$, where size $|\mathfrak{I}_b| = b$. By Lemma 8

$$\epsilon^{(b)}_{\max} - \epsilon^{(b)}_{\min} = \begin{cases} \Delta_1 \cdot Q/q^2, & \text{if } \mathcal{A} \text{ is guessed Type-1}, \\ \Delta_2 \cdot Q/q^3, & \text{if } \mathcal{A} \text{ is guessed Type-2}, \end{cases}$$

which is non-negligible and fails to lead to a good lower bound on $\left|\Pr[W_2] - \frac{1}{2}\right|$. Thus we apply Waters' approach [28] to add an artificial abort. With the strategy, $(\epsilon^{(b)}_{\max} - \epsilon^{(b)}_{\min})$ is less than $\epsilon^{(b)}_{\min} \cdot \left|\Pr[W_1] - \frac{1}{2}\right|$ and therefore $\left|\Pr[W_2] - \frac{1}{2}\right|$ is not less than

$$\frac{1}{2} \cdot \epsilon^{(b)}_{\min} \left| \Pr[W_1] - \frac{1}{2} \right| \geq \begin{cases} \frac{\left|\Pr[W_1] - \frac{1}{2}\right|}{8q|\mathcal{T}|}, & \text{if } b = 1; \\ \frac{\left|\Pr[W_1] - \frac{1}{2}\right|}{8q^2 Q^{\mathsf{id}} \cdot |\mathcal{T}|}, & \text{if } b = 2. \end{cases}$$

In summary, the advantages of $\mathcal{A}$ in Game 1 and Game 2 (no matter which type $\mathcal{A}$ is) satisfy

$$\left| \Pr[W_2] - \frac{1}{2} \right| \geq \frac{\left|\Pr[W_1] - \frac{1}{2}\right|}{8q^2 Q^{\mathsf{id}} \cdot |\mathcal{T}|}. \tag{9}$$

Game 3. We now change how $\mathbf{A}$ and $\mathbf{C}, \mathbf{G}$ in Game 2 are chosen. The game 3 challenger generates $\mathbf{A}$ as a random matrix in $\mathbb{Z}_q^{m \times n}$, but generates $\mathbf{C}, \mathbf{G}$ using algorithm $\mathsf{TrapGen}$ so that $\mathbf{C}, \mathbf{G}$ are random matrices in $\mathbb{Z}_q^{m \times n}$ and the Game 3 challenger owns trapdoors $\mathbf{T_C}, \mathbf{T_G}$ for $\Lambda_q^{\perp}(\mathbf{C}), \Lambda_q^{\perp}(\mathbf{G})$, respectively. The construction of $\mathbf{B}_i, \mathbf{D}_i$ for $i \in [\ell]$ keeps same as in Game 2, i.e. by (7). The challenger responds the queries as follows:

$\mathsf{PriKeyGen}(\cdot)$: The Game 3 challenger does the followings to respond a private key query for $\mathsf{id} = (b_1, \cdots, b_\ell) \in \{-1, 1\}^\ell$ using the trapdoor $\mathbf{T_C}$:

1. Randomly choose an unassigned leaf node ν in BT and assign it to the identity id. Save (id, ν) to UL. Run $\mathsf{SD.Assign}(\mathsf{BT}, \nu)$ to obtain $PV_\nu = \{S_{i,j}\}$.
2. Construct $\mathbf{R}^*_{\mathsf{id}} \leftarrow \sum_{i=1}^{\ell} b_i \mathbf{R}^*_i \in \mathbb{Z}_q^{m \times m}$ and $h_{\mathsf{id}} \leftarrow 1 + \sum_{i=1}^{\ell} b_i h_i$ from (7), then $\mathbf{A}_{\mathsf{id}} = \mathbf{C} + \sum_{i=1}^{\ell} b_i \mathbf{B}_i = \mathbf{A}\mathbf{R}^*_{\mathsf{id}} + h_{\mathsf{id}}\mathbf{C} \in \mathbb{Z}_q^{n \times m}$. Note that $h_{\mathsf{id}} = H_{\mathrm{ID}}(\mathsf{id})$ where H_{ID} is the hash function in $\mathcal{H}_{\mathrm{Wat}}$ defined by $(h_1, \cdots, h_\ell)$.
3. If $h_{\mathsf{id}} = 0$, abort the game and pretend that the adversary outputs a random bit r' in $\{0,1\}$, as in Game 2. Else, go to next step.
4. For each $S_{i,j} \in PV_\nu$, perform as follows:

– Retrieve the record $(\mathsf{GL}^i_{d_j}, *)$ from FL. If the second coordinate $*$ is not $\perp$, go to next step. Else, randomly select $\mathsf{F}^i_{d_j}(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $\mathsf{F}^i_{d_j}$ is a polynomial of degree 1 and $\mathsf{F}^i_{d_j}(0) = \mathbf{u}$. Update $(\mathsf{GL}^i_{d_j}, \mathsf{F}^i_{d_j})$ to FL.
– Sample $\mathbf{e}^{\mathsf{id}}_{i,j}$ as

$$\mathbf{e}^{\mathsf{id}}_{i,j} \leftarrow \mathsf{SampleRight}\left(\mathbf{A}, h_{\mathsf{id}}\mathbf{C}, \mathbf{R}^*_{\mathsf{id}}, \mathbf{T}_{\mathbf{C}}, \mathsf{F}^i_{d_j}(j), \sigma\right).$$

5. Output the updated state st and the private key $\mathsf{sk}_{\mathsf{id}}$ for id as

$$\left(PV_\nu, \{\mathbf{e}^{\mathsf{id}}_{i,j}\}_{S_{i,j} \in PV_\nu}\right).$$

$\mathsf{UpdateKey}(\cdot)$: The Game 3 challenger does the followings to respond an update key query for $\mathsf{t} = (t_1, \cdots, t_\ell) \in \{-1, 1\}^\ell$ using the trapdoor $\mathbf{T}_{\mathbf{G}}$:

1. Take R as a set consisting of such id's that for some $\mathsf{t}' \leq \mathsf{t}$, $(\mathsf{id}, \mathsf{t}') \in \mathsf{RL}$. By using UL, define RI as the set of index of leaf nodes corresponding the id's in R.
2. Run $\mathsf{SD.Cover}(\mathsf{BT}, RI)$ to obtain $CV_{\mathrm{RI}} = \{S_{i,j}\}$.
3. Construct $\mathbf{R}^*_{\mathsf{t}} \leftarrow \sum_{i=1}^{\ell} t_i \mathbf{R}^*_{\ell+i} \in \mathbb{Z}_q^{m \times m}$ and $h_{\mathsf{t}} \leftarrow 1 + \sum_{i=1}^{\ell} t_i h_{\ell+i}$ from (7), then $\mathbf{A}_{\mathsf{t}} = \mathbf{G} + \sum_{i=1}^{\ell} t_i \mathbf{D}_i = \mathbf{A}\mathbf{R}^*_{\mathsf{t}} + h_{\mathsf{t}}\mathbf{G} \in \mathbb{Z}_q^{n \times m}$. Note that $h_{\mathsf{t}} = H_{\mathrm{T}}(\mathsf{t})$ where H_{T} is the hash function in $\mathcal{H}_{\mathrm{Wat}}$ defined by $(h_{\ell+1}, \cdots, h_{2\ell})$.
4. If $h_{\mathsf{t}} = 0$, abort the game and pretend that the adversary outputs a random bit r' in $\{0, 1\}$, as in Game 2. Else, go to next step.
5. For each $S_{i,j} \in CV_{\mathrm{RI}}$, perform the following:
   – Retrieve the record $(\mathsf{GL}^i_{d_j}, *)$ from FL. If the second coordinate $*$ is not $\perp$, go to next step. Else, randomly select $\mathsf{F}^i_{d_j}(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $\mathsf{F}^i_{d_j}$ is a polynomial of degree 1 and $\mathsf{F}^i_{d_j}(0) = \mathbf{u}$. Update $(\mathsf{GL}^i_{d_j}, \mathsf{F}^i_{d_j})$ to FL.
   – Sample $\mathbf{e}^{\mathsf{t}}_{i,j}$ as

$$\mathbf{e}^{\mathsf{t}}_{i,j} \leftarrow \mathsf{SampleRight}\left(\mathbf{A}, h_{\mathsf{t}}\mathbf{G}, \mathbf{R}^*_{\mathsf{t}}, \mathbf{T}_{\mathbf{G}}, \mathsf{F}^i_{d_j}(j), \sigma\right).$$

6. Output the update key uk_{t} at time t as

$$\left(CV_{\mathrm{RI}}, \{\mathbf{e}^{\mathsf{t}}_{i,j}\}_{S_{i,j} \in CV_{\mathrm{RI}}}\right).$$

For other phases, Game 3 is identical to Game 2. Specially, in the challenge phase the challenger checks which type the adversary $\mathcal{A}$ is.

a. If $\mathcal{A}$ is Type-1, the challenger checks if the challenge time $\mathsf{t}^* = (t^*_1, \cdots, t^*_\ell) \in \{-1, 1\}^\ell$ satisfies $h_{\mathsf{t}^*} := 1 + \sum_{i=1}^{\ell} t^*_i h_{\ell+i} = 0$. If not, the challenger aborts the game as in Game 2. Similarly, an artificial abort is implemented in Game 3.
b. If $\mathcal{A}$ is Type-2, the challenger checks if the challenger time $\mathsf{t}^* = (t^*_1, \cdots, t^*_\ell) \in \{-1, 1\}^\ell$ satisfies $h_{\mathsf{t}^*} := 1 + \sum_{i=1}^{\ell} t^*_i h_{\ell+i} = 0$ and together the challenge identity $\mathsf{id}^* = (b^*_1, \cdots, b^*_\ell) \in \{-1, 1\}^\ell$ satisfies $h_{\mathsf{id}^*} := 1 + \sum_{i=1}^{\ell} b^*_i h_i = 0$. If not, the challenger aborts the game as in Game 2. Similarly, an artificial abort is implemented in Game 3.

Now we prove the outputs from $\mathsf{SampleRight}$ we used in Game 3 here are indistinguishable from the corresponding outputs from $\mathsf{SampleLeft}$ in Game 2. We consider the responses to private key queries first. Since h_{id} in Step 4 of $\mathsf{PriKeyGen}$ is non-zero, the matrix $\mathbf{T}_{\mathbf{C}}$ is also a trapdoor for $h_{\mathsf{id}}\mathbf{C}$. Moreover, $h_{\mathsf{id}}\mathbf{C}$ is a matrix with rank n with high probability as $\mathbf{C}$ is. By the introduction in Section 2.3, when $\sigma \geq \|\widetilde{\mathbf{T}_{\mathbf{C}}}\| \cdot s_{\mathbf{R}^*_{\mathsf{id}}} \cdot \omega(\sqrt{\log m})$, where $s_{\mathbf{R}^*_{\mathsf{id}}} = \|\mathbf{R}^*_{\mathsf{id}}\|_R \leq \sum_{i=1}^{\ell} \|\mathbf{R}^*_i\|_R = O(\ell\sqrt{m})$ from Lemma 4, the generated $\mathbf{e}^{\mathsf{id}}_{i,j} \in \mathbb{Z}^{2m}$ is distributed statistically close to $\mathcal{D}_{\Lambda^{\mathbf{f}}_q(\mathbf{F}_{\mathsf{id}}), \sigma}$, where $\mathbf{f} = \mathsf{F}^i_{d_j}(j)$ and $\mathbf{F}_{\mathsf{id}} = (\mathbf{A}|\mathbf{A}\mathbf{R}^*_{\mathsf{id}} + h_{\mathsf{id}}\mathbf{C}) = (\mathbf{A}|\mathbf{C} + \sum_{i=1}^{\ell} b_i \mathbf{B}_i)$, as Game 2. Recall that $\|\widetilde{\mathbf{T}_{\mathbf{C}}}\| \leq O(\sqrt{m})$ from Proposition 1. Therefore σ we select in (4) is sufficiently large to meet our sampling requirement. The situation for update key queries is similar.

Since from the adversary's view, the public parameters, responses to private key and update key queries, the challenge ciphertext and abort conditions are all indistinguishable between Game 2 and Game 3, the advantages of the adversary in both games are identical, i.e.

$$\Pr[W_2] = \Pr[W_3]. \tag{10}$$

Remark: In Game 2, we have a uniform random hash function $\tilde{H} \in \mathcal{H}_{\mathrm{Wat}}$ in common. In Game 3, we have two separate uniform random hash functions $H_{\mathrm{ID}}, H_{\mathrm{T}}$. But in terms of the non-abort probability, it is easy to show that there is no difference. We adopt one hash instead of two in Game 2 to make the description concise and precise.

Game 4. Game 4 is the same as Game 3 except that the challenge ciphertext $(c_0^*, \mathbf{c}_1^*, \mathbf{c}_2^*)$ is always chosen as a random independent element in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$. As the challenge ciphertext is randomly chosen from the ciphertext space, the advantage of the adversary is zero.

Next we show Game 3 and Game 4 are computationally indistinguishable by giving a reduction from the LWE problem.

Reduction from LWE. Suppose the adversary $\mathcal{A}$ can distinguish Game 3 and Game 4. We use $\mathcal{A}$ to construct an LWE algorithm $\mathcal{B}$. Form Definition 5, we know an LWE instance is provided as a sampling oracle $\mathcal{O}$ which can be either truly random $\mathcal{O}_{\$}$ or noisy pseudo-random $\mathcal{O}_{\mathbf{s}}$ for some secret key $\mathbf{s} \in \mathbb{Z}_q^n$. The simulator $\mathcal{B}$ utilizes the adversary $\mathcal{A}$ to distinguish between the two as follows:

Instance. $\mathcal{B}$ requests from $\mathcal{O}$ and receives a fresh pair $(\mathbf{u}_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for each $i = 0, \cdots, m$.

Setup. $\mathcal{B}$ prepares a simulated attack environment for $\mathcal{A}$ as follows:

1. Assemble the random matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ from m of the given LWE instances, that is, let $\mathbf{A}_0 = [\mathbf{u}_1 | \cdots | \mathbf{u}_m]$.
2. Construct the remainder of the public parameters, namely, $\mathbf{B}_i, \mathbf{D}_i$ for $i \in [\ell]$ and $\mathbf{C}, \mathbf{G}$ as in Game 3 using random h_i and $\mathbf{R}_i^*$ for $i \in [2\ell]$.
3. Send $\mathsf{PP} = (\mathbf{A}_0, \{\mathbf{B}_1, \cdots, \mathbf{B}_\ell, \mathbf{C}\}, \{\mathbf{D}_1, \cdots, \mathbf{D}_\ell, \mathbf{G}\}, \mathbf{u}_0)$ to $\mathcal{A}$, where $\mathbf{u}_0$ is the zeroth LWE sample.

Queries. $\mathcal{B}$ responds to private key queries and update key queries as in Game 3, including aborting the game if required.

Challenge. $\mathcal{A}$ sends a target identity $\mathsf{id}^* = (b_1^*, \cdots, b_\ell^*) \in \{-1, 1\}^\ell$, a target time $\mathsf{t}^* = (t_1^*, \cdots, t_\ell^*) \in \{-1, 1\}^\ell$ and a message bit $b^* \in \{0, 1\}$. $\mathcal{B}$ prepares the challenge ciphertext as follows:

1. Retrieve $v_0, \cdots, v_m$ from the LWE instance and set $\mathbf{v}^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m$.
2. Blind the message bit by setting $c_0^* = Dv_0 + b^* \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$.
3. Set $\mathbf{R}_{\mathsf{id}^*}^* \leftarrow \sum_{i=1}^{\ell} b_i^* \mathbf{R}_i^* \in \mathbb{Z}_q^{m \times m}$ and $\mathbf{R}_{\mathsf{t}^*}^* \leftarrow \sum_{i=1}^{\ell} t_i^* \mathbf{R}_{\ell+i}^* \in \mathbb{Z}_q^{m \times m}$ where for $i \in [2\ell]$, $\mathbf{R}_i^*$ were selected at the setup phase.
4. Set $\mathbf{c}_1^* = D \cdot \begin{bmatrix} \mathbf{v}^* \\ (\mathbf{R}_{\mathsf{id}^*}^*)^\top \mathbf{v}^* \end{bmatrix}$, $\mathbf{c}_2^* = D \cdot \begin{bmatrix} \mathbf{v}^* \\ (\mathbf{R}_{\mathsf{t}^*}^*)^\top \mathbf{v}^* \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
5. Send $\mathsf{ct}^* = (c_0^*, \mathbf{c}_1^*, \mathbf{c}_2^*) \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$ to the adversary.

We claim that when $\mathcal{O}$ is a pseudo-random LWE oracle $\mathcal{O}_{\mathbf{s}}$ then ct^* is distributed exactly as in Game 3. First observe that $\mathbf{A}_{\mathsf{id}^*} = \mathbf{A}_0 \mathbf{R}_{\mathsf{id}^*}^*$. Second, by the definition of $\mathcal{O}_{\mathbf{s}}$, $\mathbf{v}^* = \mathbf{A}_0^\top \mathbf{s} + \mathbf{y}$ for some random noise vector $\mathbf{y} \in \mathbb{Z}_q^m$ distributed as $\overline{\Psi}_\alpha^m$. Therefore,

$$\begin{aligned} \mathbf{c}_1^* &= D \cdot \begin{bmatrix} \mathbf{A}_0^\top \mathbf{s} + \mathbf{y} \\ (\mathbf{R}_{\mathsf{id}^*}^*)^\top \mathbf{A}_0^\top \mathbf{s} + (\mathbf{R}_{\mathsf{id}^*}^*)^\top \mathbf{y} \end{bmatrix} \\ &= (\mathbf{A}_0 | \mathbf{A}_{\mathsf{id}^*})^\top \cdot (D\mathbf{s}) + D \cdot \begin{bmatrix} \mathbf{y} \\ (\mathbf{R}_{\mathsf{id}^*}^*)^\top \mathbf{y} \end{bmatrix} \end{aligned}$$

and similarly

$$\mathbf{c}_2^* = (\mathbf{A}_0 | \mathbf{A}_{\mathsf{t}^*})^\top \cdot (D\mathbf{s}) + D \cdot \begin{bmatrix} \mathbf{y} \\ (\mathbf{R}_{\mathsf{t}^*}^*)^\top \mathbf{y} \end{bmatrix},$$

which implies $\mathbf{c}_1^*$ and $\mathbf{c}_2^*$ are precisely $\mathbf{c}_1, \mathbf{c}_2$ parts of a valid challenge ciphertext in Game 3. Also we have $v_0 = \mathbf{u}_0^\top \mathbf{s} + x$ for some x distributed as $\overline{\Psi}_\alpha$, and therefore $c_0^* = \mathbf{u}_0^\top \cdot (D\mathbf{s}) + Dx + b^* \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$, which is precisely c_0 part of a valid challenge ciphertext in Game 3.

When $\mathcal{O}$ is a random oracle $\mathcal{O}_\$$, then v_0 is uniform in $\mathbb{Z}_q$ and $\mathbf{v}^*$ is uniform in $\mathbb{Z}_q^m$. Therefore $(c_0^*, \mathbf{c}_1^*, \mathbf{c}_2^*)$ is uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$ by the standard leftover hash lemma (e.g Theorem 8.38 of [26]). Hence the challenge ciphertext is always uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$, as in Game 4.

Guess. $\mathcal{A}$ is allowed to make more queries, then guesses if it is interacting with a Game 3 or Game 4 challenger. Next the simulator implements the artificial abort from Game 3 and 4, then outputs the final guess to the LWE challenger.

As discussed above, if $\mathcal{O} = \mathcal{O}_{\mathbf{s}}$, the adversary's view is the same as in Game 3; if $\mathcal{O} = \mathcal{O}_\$$, the adversary's view is the same as in Game 4. Thus the advantage of $\mathcal{B}$ in solving LWE is the same as the advantage of $\mathcal{A}$ in distinguishing Game 3 and Game 4, as needed. Since $\Pr[W_4] = 1/2$, we obtain

$$\left|\Pr[W_3] - \frac{1}{2}\right| = |\Pr[W_3] - \Pr[W_4]| \leq \mathsf{Adv}_{\mathcal{B}}^{(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)\text{-LWE}}. \tag{11}$$

By combining (8)-(11), we obtain

$$\left|\Pr[W_0] - \frac{1}{2}\right| \leq 8q^2 Q^{\mathsf{id}} \cdot |\mathcal{T}| \cdot \mathsf{Adv}_{\mathcal{B}}^{(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)\text{-LWE}},$$

as stated in Theorem 1.