Server-Aided Revocable Identity-Based Encryption from Lattices

Khoa Nguyen, Huaxiong Wang, Juanyang Zhang

Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
{khoantt,hxwang,zh0078ng}@ntu.edu.sg

Abstract. Server-aided revocable identity-based encryption (SR-IBE), recently proposed by Qin et al. at ESORICS 2015, offers significant advantages over previous user revocation mechanisms in the scope of IBE. In this new system model, almost all the workloads on users are delegated to an untrusted server, and users can compute decryption keys at any time period without having to communicate with either the key generation center or the server.

In this paper, inspired by Qin et al.'s work, we design the first SR-IBE scheme from lattice assumptions. Our scheme is more efficient than existing constructions of lattice-based revocable IBE. We prove that the scheme is selectively secure in the standard model, based on the hardness of the Learning with Errors problem. At the heart of our design is a "double encryption" mechanism that enables smooth interactions between the message sender and the server, as well as between the server and the recipient, while ensuring the confidentiality of messages.

1 Introduction

Identity-based encryption (IBE), envisaged by Shamir [34] in 1984, allows to use arbitrary strings representing users' identities (e.g., email addresses) as public keys, and thus, greatly simplifies the burden of key management in traditional public-key infrastructure (PKI). In an IBE scheme, there is a trusted authority, called the Key Generation Center (KGC), who is in charge of generating a private key corresponding to each identity and sending it to the user through a secret channel. Such private key enables the user to recover messages encrypted under his identity. Shamir's ideas triggered an exciting search for provably secure IBE systems, but the first realizations only appeared in 2001, when Boneh and Franklin [8] and Cocks [13] presented constructions based on pairings and on the quadratic residual problem, respectively. The third class of IBE, pioneered by Gentry et al. [16] in 2008, is based on lattice assumptions.

As for many multi-user cryptosystems, an efficient revocation mechanism is necessary and imperative in the IBE setting. If some identities have been revoked due to certain reasons (e.g., the user misbehaves or his private key is stolen), the mechanism should ensure that: (i) the revoked identities no longer possess the decryption capability; (ii) the workloads of the KGC and the non-revoked users in updating the system are "small". Designing an IBE scheme supported by efficient revocation turned out to be a challenging problem. A naïve solution, suggested by Boneh and Franklin in their seminal work [8], requires users to periodically renew their private keys by communicating with the KGC per time epoch, via a secure channel. This solution, while yielding a straightforward revocation method (i.e., revoked identities are not given new keys), is too impractical to be used for large-scale system, as the workload of the KGC grows linearly in the number of users N. Later on, Boldyreva, Goyal and Kumar (BGK) [6] formally defined the notion of revocable identity-based encryption (RIBE), and employed the tree-based revocation techniques from [27] to construct the first scalable RIBE in which the KGC's workload is only logarithmic in N. In the BGK model, however, the non-revoked users have to communicate with the KGC regularly to receive the update keys. Although this key updating process can be done through a public channel, it is somewhat inconvenient and bandwidth-consuming.

To improve the situation, Qin et al. [30] recently proposed server-aided revocable identity-based encryption (SR-IBE) - a new revocation approach in which almost all workloads on users are outsourced to a server,

and users can compute decryption keys at any time period without having to communicate with either the KGC or the server. Moreover, the server can be untrusted (in the sense that it does not possess any secret information) and should just perform correct computations. More specifically, an SR-IBE scheme functions as follows. When setting up the system, the KGC issues a long-term private key to each user. The update keys are sent only to the server (via a public channel) rather than to all users. The ciphertexts also go through the server who transforms them to "partially decrypted ciphertexts" which are forwarded to the intended recipients. The latter then can recover the messages using decryption keys derived from their long-term keys. This is particularly well-suited for applications such as secure email systems, where email addresses represent users' identities and the (untrusted) email server performs most of the computations. In [30], apart from introducing this new model, Qin et al. also described a pairing-based instantiation of SR-IBE.

In this work, inspired by the advantages and potentials of SR-IBE, we put it into the world of lattice-based cryptography, and design the first SR-IBE scheme from lattice assumptions.

Related Works. The subset cover framework, originally proposed by Naor, Naor and Lotspiech (NNL) [27] in the context of broadcast encryption, is arguably the most well-known revocation technique for multi-user systems. It uses a binary tree, each leaf of which is designated to each user. Non-revoked users are partitioned into disjoint subsets, and are assigned keys according to the Complete Subtree (CS) method or the Subset Difference (SD) method. This framework was first considered in the IBE setting by Boldyreva et al. [6]. Subsequently, several pairing-based RIBE schemes [24,33,18] were proposed, providing various improvements. Among them, the work by Seo and Emura [33] suggested a strong security notion for RIBE, that takes into account the threat of decryption key exposure attacks. The NNL framework also found applications in the context of revocable group signatures [22,21].

The study of IBE with outsourced revocation was initiated by Li et al. [19], who introduced a method to outsource the key update workload of the trusted KGC to a semi-trusted KGC. Indeed, revocation mechanisms with an online semi-trusted third party (called mediator) had appeared in earlier works [7,14,23,5]. However, all these approaches are vulnerable against collusion attacks between revoked users and the semi-trusted KGC or the mediator.

Lattice-based cryptography has been an exciting research area since the seminal works of Regev [31] and Gentry et al. [16]. Lattices not only allow to build powerful primitives (e.g., [15,17]) that have no feasible instantiations in conventional number-theoretic cryptography, but they also provide several advantages over the latter, such as conjectured resistance against quantum adversaries and faster arithmetic operations. In the scope of lattice-based IBE and hierarchical IBE (HIBE), numerous schemes have been introduced, in the random oracle model [16,2] and the standard model [10,1,36,37]. Chen et at. [11] employed Agrawal et al.'s IBE [1] and the CS method to construct the first revocable IBE from lattices, which satisfies selective security in the standard model. The second scheme, proposed by Cheng and Zhang [12], achieves adaptive security, via the SD method. Both of these works follow the BGK model [6].

Our Results and Techniques. We introduce the first construction of lattice-based SR-IBE. We inherit the main efficiency advantage of Qin et al.'s model over the BGK model for RIBE: the system users do not have to communicate with any party to get update keys, as they are capable of computing decryption keys for any time period on their own. As for previous lattice-based RIBE schemes [11,12], our proposal works with one-bit messages, but multi-bit variants can be achieved with small overhead, using standard techniques [16,1]. The public parameters and the ciphertexts produced by the scheme have bit-sizes comparable to those of [11,12]. The long-term private key of each user has size constant in the number of all users N, but to enable the delegation of decryption keys, it has to be a trapdoor matrix with relatively large size. The full efficiency comparison among the schemes from [11,12] and ours is given in Table 1.

As a high level, our design approach is similar to the pairing-based instantiation by Qin et al., in the sense that we also employ an RIBE scheme [11] and a two-level HIBE scheme [1] as the building blocks. In our setting, the server simultaneously plays two roles: it is the decryptor in the RIBE block (i.e., it receives ciphertexts from senders and performs the decryption mechanism of RIBE - which is called "partial decryption" here), and at the same time, it is the sender in the HIBE block. The users (i.e., the message recipients), on the other hand, only work with the HIBE block. Their identities are placed at the first level

		Public Params. Size	Token Size	Private Key Size	Update Key Size	Ciphertext Size	Model
[11]		$\widetilde{O}(\lambda^2)$	–	$O(\log N) \cdot \widetilde{O}(\lambda)$	$r \log \frac{N}{r} \cdot \widetilde{O}(\lambda)$	$\widetilde{O}(\lambda)$	Selective
[12]		$\widetilde{O}(\lambda^{2+\epsilon})$	–	$O(\log^2 N) \cdot \widetilde{O}(\lambda)$	$(2r-1) \cdot \widetilde{O}(\lambda)$	$\widetilde{O}(\lambda^{1+\epsilon})$	Adaptive
Ours	Server	$\widetilde{O}(\lambda^2)$	$O(\log N) \cdot \widetilde{O}(\lambda)$	–	$r \log \frac{N}{r} \cdot \widetilde{O}(\lambda)$	$\widetilde{O}(\lambda)$	Selective
	User		–	$\widetilde{O}(\lambda^2)$	–	$\widetilde{O}(\lambda)$	

Table 1. Comparison among known lattice-based revocable IBE schemes. Here, λ is the security parameter, N is the maximum number of users, r is the number of revoked users. For the scheme from [12], the number ϵ is a small constant such that $\epsilon < 1/2$. The notation "-" means that such an item does not exist in the corresponding scheme.

of the hierarchy, while the time periods are put at the second level. This enables the user with private key for id to delegate a decryption key for an ordered pair of the form $(\mathsf{id}, \mathsf{t})$.

However, looking into the details, it is not straightforward to make the two building blocks operate together. Qin et al. address this problem by using a key splitting technique which currently seems not available in the lattice setting. Instead, we adapt a double encryption mechanism, recently employed by Libert et al. [20] in the context of lattice-based group signatures with message-dependent opening [32], which works as follows. The sender encrypts the message under the HIBE to obtain an initial ciphertext of the form $(\mathbf{c}_2, c_0)$, where c_0 is an element of $\mathbb{Z}_q$ (for some $q > 2$) and is the ciphertext component carrying the message information. Next, he encrypts the binary representation of c_0, i.e., vector $\mathsf{bin}(c_0) \in \{0,1\}^{\lceil \log q \rceil}$, under the RIBE to obtain $(\mathbf{c}_1, \hat{\mathbf{c}}_0)$. The final ciphertext is then set as $(\mathbf{c}_1, \mathbf{c}_2, \hat{\mathbf{c}}_0)$ and is sent to the server. The latter will invert the second step of the encryption mechanism to get back to the initial ciphertext $(\mathbf{c}_2, c_0)$. Receiving $(\mathbf{c}_2, c_0)$ from the server, the user should be able to recover the message.

The security of our SR-IBE scheme relies on that of the two lattice-based building blocks, i.e., Agrawal et al.'s HIBE [1] and Chen et al.'s RIBE. Both of them are selectively secure in the standard model, assuming the hardness of the Learning with Errors (LWE) problem - so is our scheme.

Organization. The rest of this paper is organized as follows. Section 2 provides definitions of SR-IBE and some background on lattice-based cryptography. Our construction of lattice-based SR-IBE and its analysis are presented in Sections 3 and 4, respectively. We summarize our results and discuss open problems in Section 5.

2 Background and Definitions

Notations. The acronym PPT stands for "probabilistic polynomial-time". We say that a function $d : \mathbb{N} \to \mathbb{R}$ is negligible, if for sufficient large $\lambda \in \mathbb{N}$, $|d(\lambda)|$ is smaller than the reciprocal of any polynomial in λ. The statistical distance of two random variables X and Y over a discrete domain Ω is defined as $\Delta(X;Y) \triangleq \frac{1}{2}\sum_{s\in\Omega} |\Pr[X=s] - \Pr[Y=s]|$. If $X(\lambda)$ and $Y(\lambda)$ are ensembles of random variables, we say that X and Y are statistically close if $d(\lambda) \triangleq \Delta(X(\lambda); Y(\lambda))$ is a negligible function of λ. For a distribution χ, we often write $x \hookleftarrow \chi$ to indicate that we sample x from χ. For a finite set Ω, the notation $x \xleftarrow{\$} \Omega$ means that x is chosen uniformly at random from Ω.

We use bold upper-case letters (e.g., $\mathbf{A}, \mathbf{B}$) to denote matrices and use bold lower-case letters (e.g., $\mathbf{x}, \mathbf{y}$) to denote column vectors. For two matrices $\mathbf{A} \in \mathbb{Z}^{n\times m}$ and $\mathbf{B} \in \mathbb{Z}^{n\times m_1}$, $[\mathbf{A}|\mathbf{B}] \in \mathbb{Z}^{n\times(m+m_1)}$ is the concatenation of the columns of $\mathbf{A}$ and $\mathbf{B}$. For a vector $\mathbf{x} \in \mathbb{Z}^n$, $||\mathbf{x}||$ denotes the Euclidean norm of $\mathbf{x}$. We use $\widetilde{\mathbf{A}}$ to denote the Gram-Schmidt orthogonalization of matrix $\mathbf{A}$, and $||\mathbf{A}||$ to denote the Euclidean norm of the longest column in $\mathbf{A}$. If n is a positive integer, $[n]$ denotes the set $\{1, .., n\}$. For $c \in \mathbb{R}$, let $\lfloor c \rceil = \lceil c - 1/2 \rceil$ denote the integer closest to c.

2.1 Server-Aided Revocable Identity-Based Encryption

We first recall the definition and security model of SR-IBE, put forward by Qin et al. [30]. A server-aided revocable identity-based encryption (SR-IBE) scheme involves 4 parties: KGC, sender, recipient, and server. Algorithms among the parties are as follows:

Sys (1^λ) is run by the KGC. It takes as input a security parameter λ and outputs the system parameters params.

Setup (params) is run by the KGC. It takes as input the system parameters params and outputs public parameters pp, a master secret key msk, a revocation list RL (initially empty), and a state st. We assume that pp is an implicit input of all other algorithms.

Token $(\mathsf{msk}, \mathsf{id}, \mathsf{st})$ is run by the KGC. It takes as input the master secret key msk, an identity id, and state st. It outputs a token τ_{id} and an updated state st. The token τ_{id} is sent to the server through a public channel.

UpdKG $(\mathsf{msk}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$ is run by the KGC. It takes as input the master secret key msk, a time t, the current revocation list RL, and state st. It outputs an update key uk_{t}, which is sent to the server through a public channel.

TranKG $(\tau_{\mathsf{id}}, \mathsf{uk}_{\mathsf{t}})$ is run by the server. It takes as input a token τ_{id} and an update key uk_{t}, and outputs a transformation key $\mathsf{tk}_{\mathsf{id},\mathsf{t}}$.

PrivKG $(\mathsf{msk}, \mathsf{id})$ is run by the KGC. It takes as input the master key msk and an identity id, and outputs a private key $\mathsf{sk}_{\mathsf{id}}$, which is sent to the recipient through a *secret* channel.

DecKG $(\mathsf{sk}_{\mathsf{id}}, \mathsf{t})$ is run by the recipient. It takes as input the private key $\mathsf{sk}_{\mathsf{id}}$ and a time t. It outputs a decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$.

Enc $(\mathsf{id}, \mathsf{t}, M)$ is run by the sender. It takes as input the recipient's identity id, a time t, and a message M. It outputs a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$, which is sent to the server.

Transform $(\mathsf{ct}_{\mathsf{id},\mathsf{t}}, \mathsf{tk}_{\mathsf{id},\mathsf{t}})$ is run by the sever. It takes as input a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$, and a transformation key $\mathsf{tk}_{\mathsf{id},\mathsf{t}}$. It outputs a partially decrypted ciphertext $\mathsf{ct}'_{\mathsf{id},\mathsf{t}}$, which is sent to the recipient through a public channel.

Dec $(\mathsf{ct}_{\mathsf{id},\mathsf{t}}, \mathsf{dk}_{\mathsf{id},\mathsf{t}})$ is run by the recipient. On input a partially decrypted ciphertext $\mathsf{ct}'_{\mathsf{id},\mathsf{t}}$ and a decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$, this algorithm outputs a message M or a symbol $\perp$.

Revoke $(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$ is run by the KGC. It takes as input an identity id to be revoked, a revocation time t, the current revocation list RL, and a state st. It outputs an updated revocation list RL.

The correctness requirement for an SR-IBE scheme states that: For any $\lambda \in \mathbb{N}$, all possible state st, and any revocation list RL, if id is not revoked on a time t, and if all parties follow the prescribed algorithms, then $\mathbf{Dec}(\mathsf{ct}_{\mathsf{id},\mathsf{t}}, \mathsf{dk}_{\mathsf{id},\mathsf{t}}) = M$.

Qin et al. [30] defined semantic security against adaptive-identity chosen plaintext attacks for SR-IBE. Here, we will consider selective-identity security - a weaker security notion suggested by Boldyreva et al. [6], in which the adversary announces the challenge identity id^* and time t^* before the execution of algorithm **Setup**.

Definition 1 (SR-sID-CPA Security). *Let $\mathcal{O}$ be the set of the following oracles:*

1. $\mathbf{Token}(\cdot)$*: On input an identity id, return a token τ_{id} by running* $\mathbf{Token}(msk, id, st)$.
2. $\mathbf{UpdKG}(\cdot)$*: On input a time t, return an update key uk_t by running* $\mathbf{UpdKG}(msk, t, RL, st)$.
3. $\mathbf{PrivKG}(\cdot)$*: On input an identity id, return a private key sk_{id} by running* $\mathbf{PrivKG}(msk, id)$.
4. $\mathbf{DecKG}(\cdot,\cdot)$*: On input an identity id and a time t, return $dk_{id,t}$ by running* $\mathbf{DecKG}(sk_{id}, t)$*, where sk_{id} is from* $\mathbf{PrivKG}(msk, id)$.
5. $\mathbf{Revoke}(\cdot,\cdot)$*: On input an identity id and a time t, update RL by running* $\mathbf{Revoke}(id, t, RL, st)$.

An SR-IBE scheme is SR-sID-CPA secure if any PPT adversary $\mathcal{A}$ has negligible advantage in the following experiment:

$\boxed{\mathsf{Exp}^{\mathsf{SR\text{-}sID\text{-}CPA}}_{\mathcal{A}}(\lambda)}$

$params \leftarrow \mathbf{Sys}(1^\lambda);\ id^*, t^* \leftarrow \mathcal{A}$

$(pp, msk, st, RL) \leftarrow \mathbf{Setup}(params)$

$M_0, M_1 \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$

$r \xleftarrow{\$} \{0,1\}$

$ct_{id^*,t^*} \leftarrow \mathbf{Enc}(id^*, t^*, M_r)$

$r' \leftarrow \mathcal{A}^{\mathcal{O}}(ct_{id^*,t^*})$

Return 1 *if* $r' = r$ *and* 0 *otherwise.*

Beyond the conditions that M_0, M_1 belong to the message space $\mathcal{M}$ and they have the same length, the following restrictions are made:

1. $\mathbf{UpdKG}(\cdot)$ *and* $\mathbf{Revoke}(\cdot,\cdot)$ *can only be queried on time that is greater than or equal to the time of all previous queries.*
2. $\mathbf{Revoke}(\cdot,\cdot)$ *can not be queried on time t if* $\mathbf{UpdKG}(\cdot)$ *has already been queried on time t.*
3. *If* $\mathbf{PrivKG}(\cdot)$ *was queried on the challenge identity id^*, then* $\mathbf{Rovoke}(\cdot,\cdot)$ *must be queried on (id^*, t) for some $t \leq t^*$.*
4. *If id^* is non-revoked at time t^*, then* $\mathbf{DecKG}(\cdot,\cdot)$ *can not be queried on (id^*, t^*).*

The advantage of $\mathcal{A}$ in the experiment is defined as:

$$Adv^{\mathsf{SR\text{-}sID\text{-}CPA}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[\mathsf{Exp}^{\mathsf{SR\text{-}sID\text{-}CPA}}_{\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right|.$$

2.2 Background on Lattices

Let n, m, and $q \geq 2$ be integers. For matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define the m-dimensional lattice:

$$\Lambda_q^\perp(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^m :\ \mathbf{A}\cdot\mathbf{x} = \mathbf{0} \bmod q\right\} \subseteq \mathbb{Z}^m.$$

For any $\mathbf{u}$ in the image of $\mathbf{A}$, define the coset $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^m :\ \mathbf{A}\cdot\mathbf{x} = \mathbf{u} \bmod q\right\}$.

Trapdoors for Lattices. A fundamental tool of lattice-based cryptography is an algorithm that generates a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that is statistically close to uniform, together with a short trapdoor basis for the associated lattice $\Lambda_q^\perp(\mathbf{A})$.

Lemma 1 ([3,4,26]). *Let $n \geq 1, q \geq 2$ and $m \geq 2n\log q$ be integers. Then, there exists a PPT algorithm* $\mathsf{TrapGen}(n, q, m)$ *that outputs a pair $(\mathbf{A}, \mathbf{T_A})$ such that $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ and $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ is a basis for $\Lambda_q^\perp(\mathbf{A})$ satisfying $\|\widetilde{\mathbf{T_A}}\| \leq O(\sqrt{n\log q})$ and $\|\mathbf{T_A}\| \leq O(n\log q)$.*

Meanwhile, there exist matrices with particular structures, that admit easy-to-compute short bases. Micciancio and Peikert [26] consider such a matrix $\mathbf{G}$, which they call *primitive matrix*.

Lemma 2 ([26,29]). *Let $q \geq 2, n \geq 1$ be integers and let $k = \lceil \log q \rceil$. Let $\mathbf{g} = (1, 2, \cdots, 2^{k-1}) \in \mathbb{Z}^k$ and $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}$. Then the lattice $\Lambda_q^\perp(\mathbf{G})$ has a known basis $\mathbf{T_G} \in \mathbb{Z}^{nk\times nk}$ with $||\widetilde{\mathbf{T_G}}|| \leq \sqrt{5}$ and $||\mathbf{T_G}|| \leq \max\{\sqrt{5}, \sqrt{k}\}$.*

We also define $\mathsf{bin} : \mathbb{Z}_q \to \{0,1\}^k$ as the function mapping w to its binary decomposition $\mathsf{bin}(w)$. Note that, for all $w \in \mathbb{Z}_q$, we have $\mathbf{g}\cdot\mathsf{bin}(w) = w$.

Discrete Gaussians over Lattices. Let Λ be a lattice in $\mathbb{Z}^m$. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any parameter $s \in \mathbb{R}_{>0}$, define $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\frac{\|\mathbf{x}-\mathbf{c}\|^2}{s^2})$ and $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda}\rho_{s,\mathbf{c}}(\mathbf{x})$. The *discrete Gaussian distribution*

over Λ with center $\mathbf{c}$ and parameter s is $\mathcal{D}_{\Lambda,s,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{y})}{\rho_{s,\mathbf{c}}(\Lambda)}$, for $\forall \mathbf{y} \in \Lambda$. If $\mathbf{c} = \mathbf{0}$, we conveniently use ρ_s and $\mathcal{D}_{\Lambda,s}$.

Sampling Algorithms. It was shown in [16,10,1] that, given a lattice $\Lambda_q^\perp(\mathbf{A})$ equipped with a short basis, one can efficiently sample short pre-images, as well as delegate an equally short basis for a super-lattice. We will employ algorithms $\mathsf{SamplePre}$, $\mathsf{SampleBasisLeft}$ and $\mathsf{SampleLeft}$ from those works, defined below.

$\mathsf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{u}, s)$: On input a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a trapdoor $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter $s \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log m})$, it outputs a vector $\mathbf{e} \in \mathbb{Z}^m$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),s}$.

$\mathsf{SampleBasisLeft}(\mathbf{A}, \mathbf{M}, \mathbf{T_A}, s)$: On input a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times m_1}$, a trapdoor $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$ and a Gaussian parameter $s \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m+m_1)})$, it outputs a basis $\mathbf{T_F}$ of $\Lambda_q^\perp(\mathbf{F})$, where $\mathbf{F} = [\mathbf{A} \,|\, \mathbf{M}] \in \mathbb{Z}_q^{n\times(m+m_1)}$, while preserving the Gram-Schmidt norm of the basis (i.e., such that $||\widetilde{\mathbf{T}}_\mathbf{F}|| = ||\widetilde{\mathbf{T}}_\mathbf{A}||$).

$\mathsf{SampleLeft}(\mathbf{A}, \mathbf{M}, \mathbf{T_A}, \mathbf{U}, s)$: On input a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times m_1}$, a trapdoor $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$, a matrix $\mathbf{U} = [\mathbf{u}_1|\ldots|\mathbf{u}_k] \in \mathbb{Z}_q^{n\times k}$, and a Gaussian parameter $s \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m+m_1)})$, it outputs a matrix $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_k] \in \mathbb{Z}^{(m+m_1)\times k}$, where for each $j = 1, \ldots, k$, the column $\mathbf{e}_j$ is sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}_j}(\mathbf{F}),s}$. Here we also define $\mathbf{F} = [\mathbf{A} \,|\, \mathbf{M}] \in \mathbb{Z}_q^{n\times(m+m_1)}$.

2.3 The LWE Problem and Its Hardness Assumption

The Learning With Errors (LWE) problem, first introduced by Regev [31], plays the central role in lattice-based cryptography.

Definition 2 (LWE). *Let $n, m \geq 1, q \geq 2$, and let χ be a probability distribution on $\mathbb{Z}$. For $\mathbf{s} \in \mathbb{Z}_q^n$, let $\mathbf{A}_{\mathbf{s},\chi}$ be the distribution obtained by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \hookleftarrow \chi$, and outputting the pair $\left(\mathbf{a}, \mathbf{a}^\top\mathbf{s} + e\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The (n, q, χ)-*LWE *problem asks to distinguish m samples chosen according to $\mathbf{A}_{\mathbf{s},\chi}$ (for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$) and m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

If q is a prime power, $B \geq \sqrt{n} \cdot \omega(\log n)$, $\gamma = O(nq/B)$, then there exists an efficient sampleable B-bounded distribution χ (i.e., χ outputs samples with norm at most B with overwhelming probability) such that (n, q, χ)-LWE is as least as hard as worst-case lattice problem SIVP_γ (see [31,28,25,26]).

Since its introduction in 2005, the LWE problem has been used in hundreds of lattice-based cryptographic constructions. In the following, we will recall 2 such schemes, which are the building blocks of our SR-IBE in Section 3.

2.4 The Agrawal-Boneh-Boyen (H)IBE Scheme

In [1], Agrawal, Boneh, and Boyen (ABB) constructed a lattice-based IBE scheme which is proven secure in the standard model, and then extended it to the hierarchical setting. In their system, the KGC possesses a short basis $\mathbf{T_B}$ for a public lattice $\Lambda_q^\perp(\mathbf{B})$, generated via algorithm $\mathsf{TrapGen}$. Each identity in the hierarchy is associated with a super-lattice of $\Lambda_q^\perp(\mathbf{B})$, a short basis of which can be delegated from $\mathbf{T_B}$ using algorithm $\mathsf{SampleBasisLeft}$. Given such a trapdoor basis, each identity can run algorithm $\mathsf{SamplePre}$ to compute a short vector that allows to decrypt ciphertexts generated via a variant of the Dual-Regev cryptosystem [16].

Let n, m, q, s be the scheme parameters and let χ be the LWE error distribution. The scheme makes use of an efficient encoding function $\mathsf{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$, that is full-rank differences (FRD). Namely, for all distinct $\mathbf{u}, \mathbf{w} \in \mathbb{Z}_q^n$, the difference $\mathsf{H}(\mathbf{u}) - \mathsf{H}(\mathbf{w})$ is a full-rank matrix in $\mathbb{Z}_q^{n\times n}$. In this work, we will employ the two-level variant of the ABB HIBE.

$\mathsf{Setup}_{\mathsf{HIBE}}$: Generate $(\mathbf{B}, \mathbf{T_B}) \leftarrow \mathsf{TrapGen}(n, q, m)$. Pick $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. Output

$$\mathsf{pp}_{\mathsf{HIBE}} = (\mathbf{B}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{v}); \quad \mathsf{msk}_{\mathsf{HIBE}} = \mathbf{T_B}.$$

$\mathbf{Extract}_{\mathsf{HIBE}}$: For an identity $\mathsf{id} \in \mathbb{Z}_q^n$ at depth 1, output the private key $\mathsf{sk}_{\mathsf{id}}$ by running

$$\mathsf{SampleBasisLeft}\left(\mathbf{B}, \mathbf{B}_1 + \mathsf{H}(\mathsf{id})\mathbf{G}, \mathbf{T}_{\mathbf{B}}, s\right).$$

$\mathbf{Derive}_{\mathsf{HIBE}}$: For an identity $\mathsf{id} = (\mathsf{id}', \mathsf{id}'') \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$ at depth 2, let $\mathsf{sk}_{\mathsf{id}'}$ be the private key of id' and $\mathbf{B}_{\mathsf{id}'} = \left[\mathbf{B} | \mathbf{B}_1 + \mathsf{H}(\mathsf{id}')\mathbf{G}\right] \in \mathbb{Z}_q^{n \times 2m}$. Output $\mathsf{sk}_{\mathsf{id}}$ by running

$$\mathsf{SampleBasisLeft}\left(\mathbf{B}_{\mathsf{id}'}, \mathbf{B}_2 + \mathsf{H}(\mathsf{id}'')\mathbf{G}, \mathsf{sk}_{\mathsf{id}'}, s\right).$$

$\mathbf{Enc}_{\mathsf{HIBE}}$: To encrypt a message bit $b \in \{0,1\}$ under an identity $\mathsf{id} = (\mathsf{id}', \mathsf{id}'') \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$ at depth 2, let $\mathbf{B}_{\mathsf{id}} = \left[\mathbf{B} | \mathbf{B}_1 + \mathsf{H}(\mathsf{id}')\mathbf{G} | \mathbf{B}_2 + \mathsf{H}(\mathsf{id}'')\mathbf{G}\right] \in \mathbb{Z}_q^{n \times 3m}$. Choose $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x} \hookleftarrow \chi^m$, $y \hookleftarrow \chi$ and $\mathbf{S}_1, \mathbf{S}_2 \xleftarrow{\$} \{-1,1\}^{m \times m}$. Set $\mathbf{c}_1 = \mathbf{B}_{\mathsf{id}}^\top \mathbf{s} + \left[\mathbf{x} | \mathbf{S}_1^\top \mathbf{x} | \mathbf{S}_2^\top \mathbf{x}\right]^\top \in \mathbb{Z}_q^{3m}$ and $c_0 = \mathbf{v}^\top \mathbf{s} + y + b \cdot \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$. Output $\mathsf{ct}_{\mathsf{id}} = (\mathbf{c}_1, c_0) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q$.
$\mathbf{Dec}_{\mathsf{HIBE}}$: Sample $\mathbf{e}_{\mathsf{id}} \leftarrow \mathsf{SamplePre}(\mathbf{B}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id}}, \mathbf{v}, s)$. Compute $d = c_0 - \mathbf{e}_{\mathsf{id}}^\top \mathbf{c}_1 \in \mathbb{Z}_q$ and output $\lfloor \frac{2}{q} d \rceil \in \{0,1\}$.

Agrawal, Boneh and Boyen showed that their scheme satisfies the notion of indistinguishability of ciphertexts under a selective-identity chosen-plaintext attack (IND-sID-CPA), proposed by Canetti et al. [9]. We restate their result in Theorem 1.

Theorem 1 (Excerpted from [1]). *The ABB HIBE scheme is IND-sID-CPA secure, provided that the (n, q, χ)-LWE assumption holds.*

2.5 Chen et al.'s RIBE Scheme

In [11], Chen et al. proposed the first RIBE scheme from lattice assumptions. Their revocation mechanism relies on the Complete Subtree (CS) method of Naor et al. [27], which was first adapted into the context of RIBE by Boldyreva et al. [6]. We will briefly recall this method.

The CS method makes use of the node selection algorithm KUNode. In the algorithm, we use the following notation: If θ is a non-leaf node, then θ_ℓ and θ_r denote the left and right child of θ, respectively. $\mathsf{Path}(\theta)$ denotes the set of nodes on the path from θ to root. Each identity id is randomly assigned to a leaf node ν_{id} and $(\nu_{\mathsf{id}}, \mathsf{t}) \in \mathsf{RL}$ if id is revoked at time t. KUNode algorithm takes as input a binary tree BT, revocation list RL and time t, and outputs a set of nodes Y. The description of KUNode is given below and an example is illustrated in Figure 1.

$\mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$
$X, Y \leftarrow \emptyset$
$\forall (\theta_i, \mathsf{t}_i) \in \mathsf{RL},$ if $\mathsf{t}_i \leq \mathsf{t},$ then add $\mathsf{Path}(\theta_i)$ to X
$\forall \theta \in X,$ if $\theta_\ell \notin X,$ then add θ_ℓ to Y; if $\theta_r \notin X,$ then add θ_r to Y
Return Y

Chen et al.'s RIBE scheme employs two instances of the ABB IBE scheme to deal with user's identity and time, respectively. To link identity to time for each tree node, the syndrome $\mathbf{u} \in \mathbb{Z}_q^n$, which is part of the public parameter, is split into two random vectors $\mathbf{u}_1, \mathbf{u}_2$ for each node. We adopt a variant of Chen et al.'s RIBE scheme, described below, to encrypt k-bit messages instead of one-bit messages.

$\mathbf{Setup}_{\mathsf{RIBE}}$: Generate $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow \mathsf{TrapGen}(n, q, m)$. Pick $\mathbf{A}_1, \mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$. Initialize the revocation list $\mathsf{RL} = \emptyset$ and let $\mathsf{st} := \mathsf{BT}$ where BT is a binary tree. Output

$$\mathsf{RL};\ \mathsf{st};\ \mathsf{pp}_{\mathsf{RIBE}} = (\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{U});\ \mathsf{msk}_{\mathsf{RIBE}} = \mathbf{T}_{\mathbf{A}}.$$

$\mathbf{PrivKG}_{\mathsf{RIBE}}$: Randomly issue an identity $\mathsf{id} \in \mathbb{Z}_q^n$ to an unassigned leaf node ν_{id} in BT. For each $\theta \in \mathsf{Path}(\nu_{\mathsf{id}})$, if $\mathbf{U}_{1,\theta}, \mathbf{U}_{21,\theta}$ are undefined, then pick $\mathbf{U}_{1,\theta} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$ and set $\mathbf{U}_{2,\theta} = \mathbf{U} - \mathbf{U}_{1,\theta}$. Return $\mathsf{sk}_{\mathsf{id}} = (\theta, \mathbf{E}_{1,\theta})_{\theta \in \mathsf{Path}(\nu_{\mathsf{id}})}$ where

$$\mathbf{E}_{1,\theta} \leftarrow \mathsf{SampleLeft}\left(\mathbf{A}, \mathbf{A}_1 + \mathsf{H}(\mathsf{id})\mathbf{G}, \mathbf{T}_{\mathbf{A}}, \mathbf{U}_{1,\theta}, s\right).$$

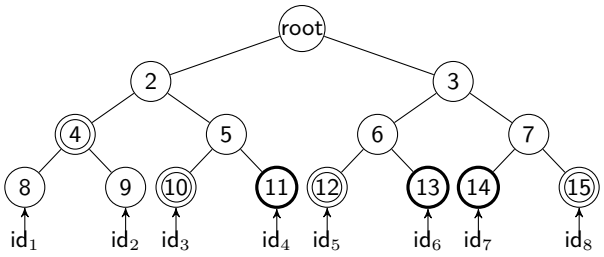

Fig. 1. Assuming that id_4, id_6 and id_7 have been revoked at time t, then $\{\theta_{11}, \theta_{13}, \theta_{14}\}$ are nodes in RL. We can get $\mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}) \rightarrow \{\theta_4, \theta_{10}, \theta_{12}, \theta_{15}\}$. For identity id_2 assigned to node θ_9, $\mathsf{Path}(\theta_9) = (\text{root} = \theta_1, \theta_2, \theta_4, \theta_9)$ and has an intersection with $\mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$ at node θ_4. For the revoked identity id_6 at node θ_{13}, $\mathsf{Path}(\theta_{13})$ does not contain any nodes in $\mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$.

$\mathbf{UpdKG}_{\mathsf{RIBE}}$: For each $\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$, retrieve $\mathbf{U}_{2,\theta}$. Output $\mathsf{uk}_\mathsf{t} = (\theta, \mathbf{E}_{2,\theta})_{\theta \in \mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$ where

$$\mathbf{E}_{2,\theta} \leftarrow \mathsf{SampleLeft}\left(\mathbf{A}, \mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}, \mathbf{T}_\mathbf{A}, \mathbf{U}_{2,\theta}, s\right).$$

$\mathbf{Enc}_{\mathsf{RIBE}}$ and **$\mathbf{Dec}_{\mathsf{RIBE}}$** are similar as in the ABB HIBE scheme. When encrypting a k-bit message, one obtains a ciphertext of the form $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathbf{c}'_1, \mathbf{c}'_0) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^k$. A non-revoked identity id at time t can obtain the pair $(\mathbf{E}_{1,\theta}, \mathbf{E}_{2,\theta})$ at the intersection node $\theta \in \mathsf{Path}(\nu_\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$, which satisfies

$$[\mathbf{A}|\mathbf{A}_1 + \mathsf{H}(\mathsf{id})\mathbf{G}] \cdot \mathbf{E}_{1,\theta} + [\mathbf{A}|\mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \cdot \mathbf{E}_{2,\theta} = \mathbf{U},$$

and which allows him to perform decryption.

$\mathbf{Revoke}_{\mathsf{RIBE}}$: Add $(\mathsf{id}, \mathsf{t})$ to RL for all nodes associated with id and return RL.

In [11], Chen et al. proved that the one-bit version of their scheme satisfies the IND-sRID-CPA security notion defined in [6], assuming the hardness of the LWE problem. The security proof can be easily adapted to handle the multi-bit case, based on the techniques from [16,1]. We thus have the following theorem.

Theorem 2 (Adapted from [11]). *The RIBE scheme described above is IND-sRID-CPA secure, provided that the (n, q, χ)-LWE assumption holds.*

3 Our Lattice-Based SR-IBE Scheme

Our SR-IBE scheme is a combination of the ABB HIBE and Chen et al.'s RIBE schemes via a double encryption technique. The KGC, who holds master secret keys for both schemes, issues HIBE private keys to users, and gives tokens consisting of RIBE private keys to the server. At each time period, the KGC sends RIBE update keys to the server. The encryption algorithm is a two-step procedure:

1. Encrypt the message M under the HIBE, with respect to an ordered pair $(\mathsf{id}, \mathsf{t})$, to obtain an initial ciphertext of the form $(\mathbf{c}_2, c_0) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q$.
2. Encrypt the binary representation $\mathsf{bin}(c_0) \in \{0,1\}^k$ of c_0, where $k = \lceil \log q \rceil$, under the RIBE, with respect to id and t, to obtain $(\mathbf{c}_1, \hat{\mathbf{c}}_0) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^k$. The final ciphertext is defined as $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathbf{c}_1, \mathbf{c}_2, \hat{\mathbf{c}}_0) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^k$.

If id is not revoked at time t, then the server can partially decrypt $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$, using a transformation key which is essentially the RIBE decryption key. Note that the "partially decrypted ciphertext" is nothing but the initial ciphertext $(\mathbf{c}_2, c_0)$. Receiving $(\mathbf{c}_2, c_0)$ from the server, the user decrypts it using a decryption key delegated from his long-term private key.

In the following, we will formally describe the scheme.

Sys(1^λ): On input security parameter λ, the KGC works as follows:

1. Set $n = O(\lambda)$, and choose $N = poly(\lambda)$ as the maximal number of users that the system will support.
2. Let $q = \widetilde{O}(n^4)$ be a prime power, and set $k = \lceil \log q \rceil, m = 2nk$. Note that parameters n, q, k specify vector $\mathbf{g}$, function $\mathsf{bin}(\cdot)$ and primitive matrix $\mathbf{G}$ (see Section 2.2).
3. Choose a Gaussian parameter $s = \widetilde{O}(\sqrt{m})$.
4. Set $B = \widetilde{O}(\sqrt{n})$ and let χ be a B-bounded distribution.
5. Select an FRD map $\mathsf{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ (see Section 2.4).
6. Let the identity space be $\mathcal{I} = \mathbb{Z}_q^n$, the time space be $\mathcal{T} \subset \mathbb{Z}_q^n$ and the message space be $\mathcal{M} = \{0, 1\}$.
7. Output $\mathsf{params} = (n, N, q, k, m, s, B, \chi, \mathsf{H}, \mathcal{I}, \mathcal{T}, \mathcal{M})$.

Setup(params): On input the system parameters params, the KGC works as follows:

1. Generate two independent pairs $(\mathbf{A}, \mathbf{T_A})$ and $(\mathbf{B}, \mathbf{T_B})$ by using $\mathsf{TrapGen}(n, q, m)$.
2. Select $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$, $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.
3. Initialize the revocation list $\mathsf{RL} = \emptyset$. Obtain a binary tree BT with at least N leaf nodes and set the state $\mathsf{st} = \mathsf{BT}$.
4. Set $\mathsf{pp} = (\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{U}, \mathbf{B}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{v})$ and $\mathsf{msk} = (\mathbf{T_A}, \mathbf{T_B})$.
5. Output $(\mathsf{pp}, \mathsf{msk}, \mathsf{RL}, \mathsf{st})$.

Token$(\mathsf{msk}, \mathsf{id}, \mathsf{st})$: On input the master secret key msk, an identity $\mathsf{id} \in \mathcal{I}$ and state st, the KGC works as follows:

1. Randomly choose an unassigned leaf node ν_{id} in BT and assign it to id.
2. For each $\theta \in \mathsf{Path}(\nu_{\mathsf{id}})$, if $\mathbf{U}_{1,\theta}, \mathbf{U}_{2,\theta}$ are undefined, then pick $\mathbf{U}_{1,\theta} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$, set $\mathbf{U}_{2,\theta} = \mathbf{U} - \mathbf{U}_{1,\theta}$ and store the pair $(\mathbf{U}_{1,\theta}, \mathbf{U}_{12,\theta})$ in node θ. Sample $\mathbf{E}_{1,\theta} \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{A}_1 + \mathsf{H}(\mathsf{id})\mathbf{G}, \mathbf{T_A}, \mathbf{U}_{1,\theta}, s)$. Let $\mathbf{A}_{\mathsf{id}} = [\mathbf{A} | \mathbf{A}_1 + \mathsf{H}(\mathsf{id})\mathbf{G}] \in \mathbb{Z}_q^{n \times 2m}$. Note that $\mathbf{E}_{1,\theta} \in \mathbb{Z}^{2m \times k}$ and $\mathbf{A}_{\mathsf{id}} \cdot \mathbf{E}_{1,\theta} = \mathbf{U}_{1,\theta}$.
3. Output the updated state st and $\tau_{\mathsf{id}} = (\theta, \mathbf{E}_{1,\theta})_{\theta \in \mathsf{Path}(\nu_{\mathsf{id}})}$.

UpdKG$(\mathsf{msk}, \mathsf{t}, \mathsf{st}, \mathsf{RL})$: On input the master secret key msk, a time $\mathsf{t} \in \mathcal{T}$, state st and the revocation list RL, the KGC works as follows:

1. For each $\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$, retrieve $\mathbf{U}_{2,\theta}$ (note that $\mathbf{U}_{2,\theta}$ is always pre-defined in the **Token** algorithm), and sample $\mathbf{E}_{2,\theta} \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{A}_2 + \mathbf{H}(\mathsf{t})\mathbf{G}, \mathbf{T_A}, \mathbf{U}_{2,\theta}, s)$. Let $\mathbf{A}_{\mathsf{t}} = [\mathbf{A} | \mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \in \mathbb{Z}_q^{n \times 2m}$. Note that $\mathbf{E}_{2,\theta} \in \mathbb{Z}^{2m \times k}$ and $\mathbf{A}_{\mathsf{t}} \cdot \mathbf{E}_{2,\theta} = \mathbf{U}_{2,\theta}$.
2. Output $\mathsf{uk}_{\mathsf{t}} = (\theta, \mathbf{E}_{2,\theta})_{\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})}$.

TranKG$(\tau_{\mathsf{id}}, \mathsf{uk}_{\mathsf{t}})$: On input a token $\tau_{\mathsf{id}} = (\theta, \mathbf{E}_{1,\theta})_{\theta \in I}$ and an update key $\mathsf{uk}_{\mathsf{t}} = (\theta, \mathbf{E}_{2,\theta})_{\theta \in J}$ for some set of nodes I, J, the server works as follows:

1. If $I \cap J = \emptyset$, output $\perp$.
2. Otherwise, choose $\theta \in I \cap J$ and output $\mathsf{tk}_{\mathsf{id},\mathsf{t}} = (\mathbf{E}_{1,\theta}, \mathbf{E}_{2,\theta})$. Note that $\mathbf{A}_{\mathsf{id}} \cdot \mathbf{E}_{1,\theta} + \mathbf{A}_{\mathsf{t}} \cdot \mathbf{E}_{2,\theta} = \mathbf{U}$.

PrivKG$(\mathsf{msk}, \mathsf{id})$: On input the master secret key msk and an identity $\mathsf{id} \in \mathcal{I}$, the KGC works as follows:

1. Sample $\mathbf{T}_{\mathsf{id}} \leftarrow \mathsf{SampleBasisLeft}(\mathbf{B}, \mathbf{B}_1 + \mathsf{H}(\mathsf{id})\mathbf{G}, \mathbf{T_B}, s)$.
2. Output $\mathsf{sk}_{\mathsf{id}} = \mathbf{T}_{\mathsf{id}} \in \mathbb{Z}^{2m \times 2m}$.

DecKG$(\mathsf{sk}_{\mathsf{id}}, \mathsf{t})$: On input a private key $\mathsf{sk}_{\mathsf{id}} = \mathbf{T}_{\mathsf{id}}$ and a time $\mathsf{t} \in \mathcal{T}$, the recipient works as follows:

1. Sample $\mathbf{e}_{\mathsf{id},\mathsf{t}} \leftarrow \mathsf{SampleLeft}(\mathbf{B}_{\mathsf{id}}, \mathbf{B}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}, \mathbf{T}_{\mathsf{id}}, \mathbf{v}, s)$ where $\mathbf{B}_{\mathsf{id}} = [\mathbf{B} | \mathbf{B}_1 + \mathsf{H}(\mathsf{id})\mathbf{G}] \in \mathbb{Z}_q^{n \times 2m}$.
2. Output $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = \mathbf{e}_{\mathsf{id},\mathsf{t}} \in \mathbb{Z}^{3m}$.

Enc$(\mathsf{id}, \mathsf{t}, b)$: On input an identity $\mathsf{id} \in \mathcal{I}$, a time $\mathsf{t} \in \mathcal{T}$ and a message $M \in \mathcal{M}$, the sender works as follows:

1. Set $\mathbf{A}_{\mathsf{id},\mathsf{t}} = [\mathbf{A} | \mathbf{A}_1 + \mathsf{H}(\mathsf{id})\mathbf{G} | \mathbf{A}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \in \mathbb{Z}_q^{n \times 3m}$ and $\mathbf{B}_{\mathsf{id},\mathsf{t}} = [\mathbf{B} | \mathbf{B}_1 + \mathsf{H}(\mathsf{id})\mathbf{G} | \mathbf{B}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \in \mathbb{Z}_q^{n \times 3m}$.
2. Sample $\mathbf{s}, \mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x}, \mathbf{x}' \hookleftarrow \chi^m$, $\mathbf{y} \hookleftarrow \chi^k$, and $y' \hookleftarrow \chi$.
3. Choose $\mathbf{R}_1, \mathbf{R}_2, \mathbf{S}_1, \mathbf{S}_2 \xleftarrow{\$} \{-1, 1\}^{m \times m}$.

4. Set $\mathbf{c}_1 = \mathbf{A}_{\mathsf{id},\mathsf{t}}^\top \mathbf{s} + \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_1^\top \mathbf{x} \\ \mathbf{R}_2^\top \mathbf{x} \end{bmatrix} \in \mathbb{Z}_q^{3m}$ and $\mathbf{c}_2 = \mathbf{B}_{\mathsf{id},\mathsf{t}}^\top \mathbf{s}' + \begin{bmatrix} \mathbf{x}' \\ \mathbf{S}_1^\top \mathbf{x}' \\ \mathbf{S}_2^\top \mathbf{x}' \end{bmatrix} \in \mathbb{Z}_q^{3m}$.
5. Compute $c_0 = \mathbf{v}^\top \mathbf{s}' + y' + M \cdot \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$, and then set $\hat{\mathbf{c}}_0 = \mathbf{U}^\top \mathbf{s} + \mathbf{y} + \mathsf{bin}(c_0) \cdot \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^k$. (Recall that $\mathsf{bin}(c_0)$ is the binary decomposition of c_0.)
6. Output $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathbf{c}_1, \mathbf{c}_2, \hat{\mathbf{c}}_0) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^k$.

Transform$(\mathsf{ct}_{\mathsf{id},\mathsf{t}}, \mathsf{tk}_{\mathsf{id},\mathsf{t}})$: On input a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathbf{c}_1, \mathbf{c}_2, \hat{\mathbf{c}}_0)$ and a transformation key $\mathsf{tk}_{\mathsf{id},\mathsf{t}} = (\mathbf{E}_1, \mathbf{E}_2)$, the server works as follows:

1. Parse $\mathbf{c}_1 = \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,1} \\ \mathbf{c}_{1,2} \end{bmatrix}$ where $\mathbf{c}_{1,i} \in \mathbb{Z}_q^m$, for $i = 0, 1, 2$. Compute $\mathbf{w} = \hat{\mathbf{c}}_0 - \mathbf{E}_1^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,1} \end{bmatrix} - \mathbf{E}_2^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,2} \end{bmatrix} \in \mathbb{Z}_q^k$.
2. Compute $\hat{c}_0' = \mathbf{g} \cdot \lfloor \frac{2}{q} \mathbf{w} \rceil \in \mathbb{Z}_q$. (Recall that $\mathbf{g} = \left(1, 2, \cdots, 2^{k-1}\right) \in \mathbb{Z}^k$.)
3. Output $\mathsf{ct}'_{\mathsf{id},\mathsf{t}} = (\mathbf{c}_2, \hat{c}_0') \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q$.

Dec$(\mathsf{ct}'_{\mathsf{id},\mathsf{t}}, \mathsf{dk}_{\mathsf{id},\mathsf{t}})$: On input a partially decrypted ciphertext $\mathsf{ct}'_{\mathsf{id},\mathsf{t}} = (\mathbf{c}_2, \hat{c}_0')$ and a decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = \mathbf{e}_{\mathsf{id},\mathsf{t}}$, the recipient works as follows:

1. Compute $w' = \hat{c}_0' - \mathbf{e}_{\mathsf{id},\mathsf{t}}^\top \mathbf{c}_2 \in \mathbb{Z}_q$.
2. Output $\lfloor \frac{2}{q} w' \rceil \in \{0, 1\}$.

Revoke$(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$: On input an identity id, a time t, the revocation list RL and state $\mathsf{st} = \mathsf{BT}$, the KGC adds $(\mathsf{id}, \mathsf{t})$ to RL for all nodes associated with identity id and returns RL.

4 Analysis

In this section, we analyze the efficiency, correctness and security of our SR-IBE scheme.

4.1 Efficiency and Correctness

Efficiency. The efficiency aspect of our SR-IBE scheme is as follows:

- The bit-size of the public parameters pp is $(6nm + nk + n) \log q = \widetilde{O}\left(\lambda^2\right)$.
- The private key $\mathsf{sk}_{\mathsf{id}}$ is a trapdoor matrix of bit-size $\widetilde{O}\left(\lambda^2\right)$.
- The bit-size of the token τ_{id} is $O(\log N) \cdot \widetilde{O}\left(\lambda\right)$.
- The update key uk_{t} has bit-size $O\left(r \log \frac{N}{r}\right) \cdot \widetilde{O}\left(\lambda\right)$.
- The ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$ has bit-size $(6m + k) \log q = \widetilde{O}\left(\lambda\right)$.
- The partially decrypted ciphertext $\mathsf{ct}'_{\mathsf{id},\mathsf{t}}$ has bit-size $(3m + 1) \log q = \widetilde{O}\left(\lambda\right)$.

Correctness. When the scheme is operated as specified, if recipient id is non-revoked at time t, then $\mathsf{tk}_{\mathsf{id},\mathsf{t}} = (\mathbf{E}_1, \mathbf{E}_2)$ satisfies that $\mathbf{A}_{\mathsf{id}} \cdot \mathbf{E}_1 + \mathbf{A}_{\mathsf{t}} \cdot \mathbf{E}_2 = \mathbf{U}$. During the **Transform** algorithm performed by the server, one has:

$$
\begin{aligned}
\mathbf{w} &= \hat{\mathbf{c}}_0 - \mathbf{E}_1^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,1} \end{bmatrix} - \mathbf{E}_2^\top \begin{bmatrix} \mathbf{c}_{1,0} \\ \mathbf{c}_{1,2} \end{bmatrix} \\
&= \mathbf{U}^\top \mathbf{s} + \mathbf{y} + \mathsf{bin}(c_0) \cdot \lfloor \frac{q}{2} \rfloor - \mathbf{E}_1^\top \left(\mathbf{A}_{\mathsf{id}}^\top \mathbf{s} + \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_1^\top \mathbf{x} \end{bmatrix} \right) - \mathbf{E}_2^\top \left(\mathbf{A}_{\mathsf{t}}^\top \mathbf{s} + \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_2^\top \mathbf{x} \end{bmatrix} \right) \\
&= \mathsf{bin}(c_0) \cdot \lfloor \frac{q}{2} \rfloor + \underbrace{\mathbf{y} - \mathbf{E}_1^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_1^\top \mathbf{x} \end{bmatrix} - \mathbf{E}_2^\top \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_2^\top \mathbf{x} \end{bmatrix}}_{\mathsf{error}}.
\end{aligned}
$$

Note that if the error term above is bounded by $q/5$, i.e., $||\mathsf{error}||_\infty < q/5$, then in Step 2 of the **Transform** algorithm, one has that $\lfloor \frac{2}{q} \mathbf{w} \rceil = \mathsf{bin}(c_0)$ which implies $\hat{c}_0' = \mathbf{g} \cdot \lfloor \frac{2}{q} \mathbf{w} \rceil = c_0$.

Then, in the **Dec** algorithm run by the recipient, one has:

$$\begin{aligned} w' &= \hat{c}'_0 - \mathbf{e}_{\mathsf{id},\mathsf{t}}^{\top}\mathbf{c}_2 \\ &= \mathbf{v}^{\top}\mathbf{s}' + y' + M \cdot \lfloor \frac{q}{2} \rfloor - \mathbf{e}_{\mathsf{id},\mathsf{t}}^{\top} \left(\mathbf{B}_{\mathsf{id},\mathsf{t}}^{\top}\mathbf{s}' + \begin{bmatrix} \mathbf{x}' \\ \mathbf{S}_1^{\top}\mathbf{x}' \\ \mathbf{S}_2^{\top}\mathbf{x}' \end{bmatrix} \right) \\ &= M \cdot \lfloor \frac{q}{2} \rfloor + \underbrace{y' - \mathbf{e}_{\mathsf{id},\mathsf{t}}^{\top} \begin{bmatrix} \mathbf{x}' \\ \mathbf{S}_1^{\top}\mathbf{x}' \\ \mathbf{S}_2^{\top}\mathbf{x}' \end{bmatrix}}_{\mathsf{error}'}. \end{aligned}$$

Similarly, if the error term is less than $q/5$, i.e., $|\mathsf{error}'| < q/5$, then the recipient should be able to recover the plaintext.

As in [1,11], the two error terms above are both bounded by $sm^2B \cdot \omega(\log n) = \widetilde{O}\left(n^3\right)$, which is much smaller than $q/5$, as we set $q = \widetilde{O}\left(n^4\right)$. This implies the correctness of our scheme.

4.2 Security Analysis

In the following theorem, we prove the selective security of our SR-IBE scheme in the standard model.

Theorem 3. *The SR-IBE scheme described in Section 3 is SR-sID-CPA secure, provided that the (n, q, χ)-LWE assumption holds.*

Proof. We will demonstrate that if there is a PPT adversary $\mathcal{A}$ succeeding in breaking the SR-sID-CPA security of our SR-IBE scheme, then we can use it to construct a PPT algorithm $\mathcal{S}$ breaking either the IND-sRID-CPA security of Chen et al.'s RIBE scheme or the IND-sID-CPA security of the ABB HIBE scheme. The theorem then follows from the facts that the two building blocks are both secure under the (n, q, χ)-LWE assumption (see Theorem 1 and Theorem 2).

Let id^* be the challenge identity and t^* be the challenge time. We consider two types of adversaries as follows.

Type I Adversary: The adversary issues a query to the private key oracle $\mathbf{PrivKG}(\cdot)$ on the challenge identity id^*. In this case, the challenge identity id^* must be revoked before the challenge time t^*.

Type II Adversary: The adversary never issues a query to the private key oracle $\mathbf{PrivKG}(\cdot)$ on the challenge identity id^*. Nevertheless, it may query the decryption key oracle $\mathbf{DecKG}(\cdot, \cdot)$ on $(\mathsf{id}^*, \mathsf{t})$ as long as $\mathsf{t} \neq \mathsf{t}^*$.

Algorithm $\mathcal{S}$ begins by randomly guessing the type of adversaries it is going to deal with. We separately describe the $\mathcal{S}$'s progress for the two types of adversaries.

Lemma 3. *If there is a PPT Type I adversary $\mathcal{A}$ breaking the SR-sID-CPA security of the SR-IBE scheme from Section 3 with advantage ϵ, then there is a PPT algorithm $\mathcal{S}$ breaking the IND-sRID-CPA security of Chen et al.'s RIBE scheme with the same advantage.*

Proof. Let $\mathcal{B}$ be the challenger in the IND-sRID-CPA game for Chen et al.'s RIBE scheme. Algorithm $\mathcal{S}$ interacts with $\mathcal{A}$ and $\mathcal{B}$ as follows.

Initial: $\mathcal{S}$ runs algorithm $\mathbf{Sys}\left(1^{\lambda}\right)$ to output $\mathsf{params} = (n, N, q, k, m, s, B, \chi, \mathsf{H}, \mathcal{I}, \mathcal{T}, \mathcal{M})$. The adversary $\mathcal{A}$ announces to $\mathcal{S}$ the target id^* and t^*, and $\mathcal{S}$ forwards them to the RIBE challenger $\mathcal{B}$.

Setup: $\mathcal{S}$ sets an empty revocation list RL and a binary tree BT as the sate st. Then $\mathcal{S}$ prepares the public parameters as follows:

1. Receive $\mathsf{pp}_{\mathsf{RIBE}} = (\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{U})$ from $\mathcal{B}$, where $\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2 \in \mathbb{Z}_q^{n \times m}, \mathbf{U} \in \mathbb{Z}_q^{n \times k}$.

2. Generate $(\mathbf{B}, \mathbf{T_B})$ by running $\mathsf{TrapGen}(n, q, m)$. Select $\mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. Let $\mathsf{pp} = (\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{U}, \mathbf{B}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{v})$, and send pp to the adversary $\mathcal{A}$. Note that the distribution of pp is exactly the one expected by $\mathcal{A}$.

Token and Update Key Oracles: If $\mathcal{A}$ queries a token for identity id, algorithm $\mathcal{S}$ forwards id to $\mathcal{B}$. Receiving an RIBE private key from $\mathcal{B}$, algorithm $\mathcal{S}$ sets it as τ_{id} and forwards to $\mathcal{A}$. Similarly, when $\mathcal{A}$ queries an update key for time t, algorithm $\mathcal{S}$ sets $\mathsf{uk_t}$ as the RIBE update key it gets by interacting with $\mathcal{B}$. Recall that for Type I adversaries, the challenge id^* must be revoked before the challenge time t^*, which means that $\mathcal{A}$ is allowed to query token for id^* and also update key for t^*.

Private Key and Decryption Key Oracles: As $\mathcal{S}$ knowns the master secret key part $\mathbf{T_B}$, it can answer all private key and decryption key queries exactly as in the real scheme.

Challenge: $\mathcal{A}$ gives two messages $M_0, M_1 \in \mathcal{M}$ to $\mathcal{S}$ who prepares the challenge ciphertext as follows:

1. Choose $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x}' \longleftarrow \chi^m$ and $y' \longleftarrow \chi$. Choose $\mathbf{S}_1, \mathbf{S}_2 \xleftarrow{\$} \{-1, 1\}^{m \times m}$.
2. Compute $\mathbf{B}_{\mathsf{id}^*, \mathsf{t}^*} = [\mathbf{B} | \mathbf{B}_1 + \mathsf{H}(\mathsf{id}^*)\mathbf{G} | \mathbf{B}_2 + \mathsf{H}(\mathsf{t}^*)\mathbf{G}]$ and set $\mathbf{c}_2^* = \mathbf{B}_{\mathsf{id}^*, \mathsf{t}^*}^\top \mathbf{s}' + \begin{bmatrix} \mathbf{x}' \\ \mathbf{S}_1^\top \mathbf{x}' \\ \mathbf{S}_2^\top \mathbf{x}' \end{bmatrix} \in \mathbb{Z}_q^{3m}$.
3. Compute $M_0' = \mathbf{v}^\top \mathbf{s}' + \mathbf{y}' + M_0 \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$ and $M_1' = \mathbf{v}^\top \mathbf{s}' + \mathbf{y}' + M_1 \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$.
4. Pick $d \xleftarrow{\$} \{0, 1\}$ and set

$$M_0'' = \mathsf{bin}(M_d') \in \{0,1\}^k; \;\; M_1'' = \mathsf{bin}(M_{1 \oplus d}') \in \{0,1\}^k,$$

where $\oplus$ denotes the addition modulo 2.
Then forward M_0'', M_1'' as two challenge messages to the RIBE challenger $\mathcal{B}$. The latter will return an RIBE ciphertext $(\mathbf{c}_1', \mathbf{c}_0') \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^k$ of M_c'' under identity id^* and time t^*, where $c \xleftarrow{\$} \{0, 1\}$.
5. Set $\mathbf{c}_1^* = \mathbf{c}_1', \hat{\mathbf{c}}_0^* = \mathbf{c}_0'$ and send the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*, \hat{\mathbf{c}}_0^*) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^k$ to $\mathcal{A}$. Note that, by construction, we have $M_c'' = \mathsf{bin}(M_{d \oplus c}')$, and thus, $(\mathbf{c}_1^*, \mathbf{c}_2^*, \hat{\mathbf{c}}_0^*)$ is an SR-IBE encryption of the message $M_{d \oplus c}$ under $(\mathsf{id}^*, \mathsf{t}^*)$. Note also that, the bit $d \oplus c$ is uniformly distributed in $\{0, 1\}$.

Guess: After being allowed to make additional queries, $\mathcal{A}$ outputs $d' \in \{0, 1\}$, which is the guess that the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*, \hat{\mathbf{c}}_0^*)$ is an encryption of $M_{d'}$. Then $\mathcal{S}$ computes $c' = d \oplus d'$ and returns it to $\mathcal{B}$ as the guess for the bit c chosen by the latter.

Recall that we assume that $\mathcal{A}$ breaks the $\mathsf{SR\text{-}sID\text{-}CPA}$ security of our SR-IBE scheme with advantage ϵ, which means

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SR\text{-}sID\text{-}CPA}}(\lambda) = \left| \Pr[d' = d \oplus c] - \frac{1}{2} \right| = \epsilon.$$

On the other hand, by construction, we have $d' = d \oplus c \Leftrightarrow d' \oplus d = c \Leftrightarrow c' = c$. It then follows that:

$$\mathsf{Adv}_{\mathcal{S}, \mathsf{RIBE}}^{\mathsf{IND\text{-}sRID\text{-}CPA}}(\lambda) = \left| \Pr[c = c'] - \frac{1}{2} \right| = \epsilon.$$

□

Lemma 4. *If there is a PPT Type II adversary $\mathcal{A}$ breaking the $\mathsf{SR\text{-}sID\text{-}CPA}$ security of our SR-IBE scheme with advantage ϵ, then there is a PPT adversary $\mathcal{S}$ breaking the $\mathsf{IND\text{-}sID\text{-}CPA}$ security of the ABB HIBE scheme with the same advantage.*

Proof. We proceed in a similar manner as in the previous lemma. Let $\mathcal{B}$ be the challenger in the $\mathsf{IND\text{-}sID\text{-}CPA}$ game for the ABB HIBE scheme. Algorithm $\mathcal{S}$ interacts with $\mathcal{A}$ and $\mathcal{B}$ as follows.

Initial: $\mathcal{S}$ first runs $\mathbf{Sys}\left(1^\lambda\right)$ to output $\mathsf{params} = (n, N, q, k, m, s, B, \chi, \mathsf{H}, \mathcal{I}, \mathcal{T}, \mathcal{M})$. Then $\mathcal{A}$ announces to $\mathcal{S}$ the target identity id^* and time t^*, and $\mathcal{S}$ forwards $(\mathsf{id}^*, \mathsf{t}^*)$ to the HIBE challenger $\mathcal{B}$.

Setup: $\mathcal{S}$ sets an empty revocation list RL and a binary tree BT as the sate st. Then $\mathcal{S}$ prepares the public parameters as follows:

1. Receive $\mathsf{pp}_{\mathsf{HIBE}} = (\mathbf{B}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{v})$ from $\mathcal{B}$ where $\mathbf{B}, \mathbf{B}_1, \mathbf{B}_2 \in \mathbb{Z}_q^{n \times m}, \mathbf{v} \in \mathbb{Z}_q^n$.
2. Generate $(\mathbf{A}, \mathbf{T}_{\mathbf{A}})$ by running $\mathsf{TrapGen}(n, q, m)$. Select $\mathbf{A}_1, \mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$.
3. Let the public parameters be $\mathsf{pp} = (\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{U}, \mathbf{B}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{v})$ and send pp to the adversary $\mathcal{A}$.

Token and Update Key Oracles: As $\mathcal{S}$ knows the master secret key part $\mathbf{T}_{\mathbf{A}}$, it can answer all token and update key queries.

Private Key and Decryption Key Oracles: When $\mathcal{A}$ issues a private key query for an identity id where $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{S}$ forwards id to $\mathcal{B}$. The answer received from the latter is then set as $\mathsf{sk}_{\mathsf{id}}$ and is sent to $\mathcal{A}$. When $\mathcal{A}$ queries a decryption key for an identity id at a time t, where $(\mathsf{id}, \mathsf{t}) \neq (\mathsf{id}^*, \mathsf{t}^*)$, algorithm $\mathcal{S}$ forwards $(\mathsf{id}, \mathsf{t}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$ to $\mathcal{B}$, and obtains a matrix $\mathbf{E} \in \mathbb{Z}^{3m \times 3m}$, which is a short trapdoor for $\Lambda_q^{\perp}(\mathbf{B}_{\mathsf{id},\mathsf{t}})$ where $\mathbf{B}_{\mathsf{id},\mathsf{t}} = [\mathbf{B} | \mathbf{B}_1 + \mathsf{H}(\mathsf{id})\mathbf{G} | \mathbf{B}_2 + \mathsf{H}(\mathsf{t})\mathbf{G}] \in \mathbb{Z}_q^{n \times 3m}$. Then $\mathcal{S}$ samples $\mathbf{e}_{\mathsf{id},\mathsf{t}} \leftarrow \mathsf{SamplePre}(\mathbf{B}_{\mathsf{id},\mathsf{t}}, \mathbf{E}, \mathbf{v}, s)$, sets $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = \mathbf{e}_{\mathsf{id},\mathsf{t}}$, and sends $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$ to $\mathcal{A}$. Note that, if $\mathsf{t} \neq \mathsf{t}^*$ then id can be the same as id^*.

Challenge: $\mathcal{A}$ gives two messages $M_0, M_1 \in \{0,1\}$ to $\mathcal{S}$, who prepares the challenge ciphertext as follows:

1. Choose $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x} \longleftarrow \chi^m$ and $\mathbf{y} \longleftarrow \chi^k$. Choose $\mathbf{R}_1, \mathbf{R}_2 \xleftarrow{\$} \{-1, 1\}^{m \times m}$.
2. Set $\mathbf{c}_1^* = \mathbf{A}_{\mathsf{id}^*,\mathsf{t}^*}^{\top}\mathbf{s} + \begin{bmatrix} \mathbf{x} \\ \mathbf{R}_1^{\top}\mathbf{x} \\ \mathbf{R}_2^{\top}\mathbf{x} \end{bmatrix} \in \mathbb{Z}_q^{3m}$ where $\mathbf{A}_{\mathsf{id}^*,\mathsf{t}^*} = [\mathbf{A} | \mathbf{A}_1 + \mathsf{H}(\mathsf{id}^*)\mathbf{G} | \mathbf{A}_2 + \mathsf{H}(\mathsf{t}^*)\mathbf{G}]$.
3. Pick $d \xleftarrow{\$} \{0,1\}$. Set $M_0' = M_d$ and $M_1' = M_{1 \oplus d}$. Forward M_0', M_1' as two challenge messages to the HIBE challenger $\mathcal{B}$. The latter will return a ciphertext $(\mathbf{c}_1'', c_0'') \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q$, which is an HIBE encryption of message M_c' under "identity" $(\mathsf{id}^*, \mathsf{t}^*)$, where $c \xleftarrow{\$} \{0,1\}$.
4. Set $\mathbf{c}_2^* = \mathbf{c}_1''$ and $\hat{\mathbf{c}}_0^* = \mathbf{U}^{\top}\mathbf{s} + \mathbf{y} + \mathsf{bin}(c_0'') \cdot \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q^k$.
5. Send $(\mathbf{c}_1^*, \mathbf{c}_2^*, \hat{\mathbf{c}}_0^*) \in \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^k$ to the adversary $\mathcal{A}$. Note that $(\mathbf{c}_1^*, \mathbf{c}_2^*, \hat{\mathbf{c}}_0^*)$ is an SR-IBE encryption of the message $M_{c \oplus d} = M_c'$ under identity id^* and time t^*.

Guess: After being allowed to make additional queries, $\mathcal{A}$ outputs $d' \in \{0,1\}$, which is the guess that the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*, \hat{\mathbf{c}}_0^*)$ is an encryption of $M_{d'}$. Then $\mathcal{S}$ computes $c' = d \oplus d'$ and returns it to $\mathcal{B}$ as the guess for the bit c chosen by the latter.

Recall that we assume that $\mathcal{A}$ breaks the $\mathsf{SR\text{-}sID\text{-}CPA}$ security of our SR-IBE scheme with probability ϵ, which means

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SR\text{-}sID\text{-}CPA}}(\lambda) = \left| \Pr[d' = d \oplus c] - \frac{1}{2} \right| = \epsilon.$$

On the other hand, by construction, we have $d' = d \oplus c \Leftrightarrow d' \oplus d = c \Leftrightarrow c' = c$. It then follows that:

$$\mathsf{Adv}_{\mathcal{S},\mathsf{HIBE}}^{\mathsf{IND\text{-}sID\text{-}CPA}}(\lambda) = \left| \Pr[c = c'] - \frac{1}{2} \right| = \epsilon.$$

□

Finally, recall that algorithm $\mathcal{S}$ can guess the type of the adversary correctly with probability $1/2$ and the adversary's behaviour is independent from the guess. It then follows from the results of Lemma 3 and Lemma 4 that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SR\text{-}sID\text{-}CPA}}(\lambda) = \frac{1}{2}\Big(\mathsf{Adv}_{\mathcal{S},\mathsf{RIBE}}^{\mathsf{IND\text{-}sRID\text{-}CPA}}(\lambda) + \mathsf{Adv}_{\mathcal{S},\mathsf{HIBE}}^{\mathsf{IND\text{-}sID\text{-}CPA}}(\lambda)\Big).$$

By Theorem 1 and Theorem 2, we then have that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SR\text{-}sID\text{-}CPA}}(\lambda) = \mathsf{negl}(\lambda)$, provided that the (n, q, χ)-LWE assumption holds. This concludes the proof. □

5 Conclusion and Open Problems

We present the first server-aided RIBE from lattice assumptions. In comparison with previous lattice-based realizations [11,12] of RIBE, our scheme has a noticeable advantage in terms of computation and communication costs on the user side. The scheme only satisfies the weak notion of selective security. Nevertheless, adaptive security in the standard model can possibly be achieved (at the cost of efficiency) by replacing the two building blocks by adaptively-secure lattice-based constructions, e.g., the RIBE from [12] and the HIBE schemes from [35,37]. One limitation of the scheme is the large size of user's long-term secret key: while being independent of the number of users, it is quadratic in the security parameter λ. Reducing this key size (e.g., making it linear in λ) is left as an open question.

Another question that we left unsolved is how to construct a lattice-based scheme secure against decryption key exposure attacks considered by Seo and Emura [33]. Existing pairing-based RIBE schemes satisfying this strong notion all employ a randomization technique in the decryption key generation procedure, that seems hard to adapt into the lattice setting. Finally, it is worth investigating whether our design approach (i.e., using a double encryption mechanism with an RIBE and an HIBE that have suitable plaintext/ciphertext spaces) would yield a generic construction for SR-IBE.

Acknowledgements. We thank Baodong Qin, Sanjay Bhattacherjee, and the anonymous reviewers for helpful discussions and comments. The research was supported by the "Singapore Ministry of Education under Research Grant MOE2013-T2-1-041". Huaxiong Wang was also supported by NTU under Tier 1 grant RG143/14.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer (2010)
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer (2010)
3. Ajtai, M.: Generating hard instances of the short basis problem. In: ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer (1999)
4. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory of Computing Systems 48(3), 535–553 (2011)
5. Baek, J., Zheng, Y.: Identity-based threshold decryption. In: PKC 2004. LNCS, vol. 2947, pp. 262–276. Springer (2004)
6. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: CCS 2008. pp. 417–426. ACM (2008)
7. Boneh, D., Ding, X., Tsudik, G., Wong, C.: A method for fast revocation of public key certificates and security capabilities. In: 10th USENIX Security Symposium. pp. 297–310. USENIX (2001)
8. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer (2001)
9. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer (2003)
10. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer (2010)
11. Chen, J., Lim, H.W., Ling, S., Wang, H., Nguyen, K.: Revocable identity-based encryption from lattices. In: ACISP 2012. LNCS, vol. 7372, pp. 390–403. Springer (2012)
12. Cheng, S., Zhang, J.: Adaptive-ID secure revocable identity-based encryption from lattices via subset difference method. In: ISPEC 2015. LNCS, vol. 9065, pp. 283–297. Springer (2015)
13. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Cryptography and Coding, 8th IMA International Conference. LNCS, vol. 2260, pp. 360–363. Springer (2001)
14. Ding, X., Tsudik, G.: Simple identity-based cryptography with mediated RSA. In: CT-RSA 2003. LNCS, vol. 2612, pp. 193–210. Springer (2003)
15. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 2009. pp. 169–178. ACM (2009)

16. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206. ACM (2008)
17. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: CRYPTO 2015. pp. 503–523. No. 9216 in LNCS (2015)
18. Lee, K., Lee, D.H., Park, J.H.: Efficient revocable identity-based encryption via subset sifference methods. Cryptology ePrint Archive, Report 2014/132 (2014), http://eprint.iacr.org/2014/132
19. Li, J., Li, J., Chen, X., Jia, C., Lou, W.: Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans. Computers 64(2), 425–437 (2015)
20. Libert, B., Mouhartem, F., Nguyen, K.: A lattice-based group signature scheme with message-dependent opening. In: ACNS 2016. LNCS, vol. 9696, pp. 137–155. Springer (2016)
21. Libert, B., Peters, T., Yung, M.: Group signatures with almost-for-free revocation. In: CRYPTO 2012. LNCS, vol. 7417, pp. 571–589. Springer (2012)
22. Libert, B., Peters, T., Yung, M.: Scalable group signatures with revocation. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 609–627. Springer (2012)
23. Libert, B., Quisquater, J.: Efficient revocation and threshold pairing based cryptosystems. In: ACM Symposium on Principles of Distributed Computing, PODC 2003. pp. 163–171. ACM (2003)
24. Libert, B., Vergnaud, D.: Adaptive-ID secure revocable identity-based encryption. In: CT-RSA 2009. LNCS, vol. 5473, pp. 1–15. Springer (2009)
25. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer (2011)
26. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer (2012)
27. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer (2001)
28. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009. pp. 333–342. ACM (2009)
29. Peikert, C.: A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science 10(4), 283–424 (2016)
30. Qin, B., Deng, R.H., Li, Y., Liu, S.: Server-aided revocable identity-based encryption. In: ESORICS 2015. LNCS, vol. 9326, pp. 286–304. Springer (2015)
31. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005. pp. 84–93. ACM (2005)
32. Sakai, Y., Emura, K., Hanaoka, G., Kawai, Y., Matsuda, T., Omote, K.: Group signatures with message-dependent opening. In: Pairing 2012. LNCS, vol. 7708, pp. 270–294. Springer (2012)
33. Seo, J.H., Emura, K.: Revocable identity-based encryption revisited: ecurity model and construction. In: PKC 2013. LNCS, vol. 7778, pp. 216–234. Springer (2013)
34. Shamir, A.: Identity-based cryptosystems and signature schemes. In: CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer (1985)
35. Singh, K., Rangan, C.P., Banerjee, A.K.: Adaptively secure efficient lattice (H)IBE in standard model with short public parameters. In: SPACE 2012. LNCS, vol. 7644, pp. 153–172. Springer (2012)
36. Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: EUROCRYPT 2016. LNCS, vol. 9666, pp. 32–62. Springer (2016)
37. Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: Short signatures and ibes with small key sizes. In: CRYPTO 2016. LNCS, vol. 9816, pp. 303–332. Springer (2016)