Revocable Identity-Based Encryption with Server-Aided Ciphertext Evolution from Lattices

Yanhua Zhang[1](✉), Ximeng Liu[2], Yupu Hu[3], and Huiwen Jia[4]

[1] Zhengzhou University of Light Industry, Zhengzhou, 450002, China
yhzhang@email.zzuli.edu.cn
[2] Fuzhou University, Fuzhou, 350108, China
snbnix@gmail.com
[3] Xidian University, Xi'an, 710071, China
yphu@mail.xidian.edu.cn
[4] Guangzhou University, Guangzhou, 510006, China
hwjia@gzhu.edu.cn

Abstract. Revocable identity-based encryption (RIBE) with server-aided ciphertext evolution (RIBE-CE), recently proposed by Sun et al. at TCS 2020, offers significant advantages over previous identity (or key) revocation mechanisms when considering the scenario of a secure data sharing in the cloud setting. In this new system model, the user (i.e., a recipient) can utilize current short-term decryption key to decrypt all ciphertexts sent to him, meanwhile, the ciphertexts in the cloud evolve to new ones with the aided of the cloud server and the old ones are completely deleted, and thus, the revoked users cannot access to both the previously and subsequently shared data.

In this paper, inspired by Sun et al.'s work, we propose the first lattice-based RIBE-CE. Our scheme is more efficient and secure than the existing constructions of lattice-based RIBE. Simultaneously, the private key generator (PKG) maintains a binary tree (BT) to handle key revocation only with a logarithmic complexity workload in time key update, not growing linearly in the numbers of system users N, which serves as one solution to the challenge proposed by Sun et al., and based on the hardness of the learning with errors (LWE) problem, we prove that our first scheme is selectively secure in the standard model. Subsequently, based on the main techniques for lattice basis delegation with hierarchical IBE (HIBE), we construct our second lattice-based RIBE-CE scheme with decryption key exposure resistance (DKER), a default security requirement for RIBE, which has not been considered by Sun et al..

Keywords: IBE · Lattices · Key revocation · Ciphertext evolution.

1 Introduction

Identity-based encryption (IBE), a seminal notion envisaged by Shamir [25] at Crypto 1984, can eliminate the needs for providing a public-key infrastructure in conventional public-key cryptosystems. Until 2001, the first realizations of IBE

based on pairings and on quadratic residual problem were introduced by Boneh and Franklin [5] and Cocks [8], respectively. In addition, Boneh and Franklin [5] also suggested a naive solution to the issue of identity (or key) revocation in IBE, that is, each non-revoked user needs to be periodically reassigned a private key by communicating with the private key generator (PKG) per time epoch via a *secret* channel. Obviously, this solution is inefficient for a large-scale IBE system, because the PKG's workload grows linearly in the number of system users N.

The first scalable IBE with key revocation, or simply revocable IBE (RIBE) was set forth by Boldyreva et al. [4] at CCS 2008, whose scheme is designed by adopting a binary tree (BT) based revocation method [19] and PKG's workload is only logarithmic in N. Though the time key updating process of [4] can be exactly executed for all the non-revoked users over a *public* channel, each non-revoked user still requires different time keys to accomplish ciphertexts decryption for different time periods, and each user in IBE should store a series of time update keys which grow linearly in the whole periods. Additionally, when considering a practical application of RIBE, there is a problem that ciphertexts generated for a user, but prior to the user's revocation, remain available to the revoked user. This problem may be undesirable for some applications, such as the scenario of a secure data sharing in the cloud setting.

To solve both the aforementioned two problems simultaneously in a practical manner, Sun et al. [27] recently introduced revocable identity-based encryption with server-aided ciphertext evolution (RIBE-CE) - a new revocation method in which the user (i.e., a recipient) has to utilize a current short-term decryption key to decrypt all ciphertexts sent to him, meanwhile, the ciphertexts in the cloud evolve to new ones with the aided of the cloud server and old ones are completely deleted, and thus, the revoked users cannot access to both the previously and subsequently shared data. To be more specific, an RIBE-CE scheme should be carried out as follows: once the system is set up, PKG issues a long-term private key to user. A time update key is generated by PKG and sent to the cloud server (and all users) via a *public* channel at each time period. The cloud server should do ciphertext evolution on the encrypted data which may be just uploaded by a data owner or have been stored in the cloud for some time to new ciphertexts by using the time update key, and the old ciphertexts are deleted. Because only the non-revoked user can obtain a valid short-term decryption key, any revoked users cannot decrypt the ciphertexts (including the former ciphertexts) sent to him. In [27], apart from introducing this new RIBE model, Sun et al. also described a generic construction and the pairing-based concrete constructions of RIBE-CE.

In this paper, inspired by the clear advantages of RIBE-CE, we bring it into the world of lattice-based cryptograph which has faster arithmetic operations and is believed to the most promising candidate for post-quantum cryptography.

Related Works. The first scalable RIBE scheme was introduced by Boldyreva et al. [4], whose scheme is constructed by combining a fuzzy IBE scheme [23] and a subset cover framework [19]. Subsequently, an adaptive-id secure RIBE and RIBE with decryption key exposure resistance (DKER) from bilinear groups were proposed by Libert and Vergnaud [15] and Seo and Emura [24], respectively.

Lattice-based cryptography, believed to be secure in a quantum computer attack environment, enjoys some competive advantages over conventional number-theoretic cryptography, such as simpler arithmetic operations and proven secure based on the *worst-case* hardness assumptions. Following the model of [4], the first lattice-based RIBE scheme without DKER, the first schemes with bounded DKER and unbounded DKER and an adaptively secure scheme in the quantum random oracle model were proposed by Chen et al. [7], Takayasu and Watanabe [29], Katsumata et al. [11] and Takayasu [28], respectively.

Cloud-based technology, including cloud computing and cloud storage, etc., has already created a new generation of computing paradigm, and with a flexible assistance of cloud (e.g., irrespective of time and location), many conventional costly computations and bulky storages can be performed with ease. Therefore, introducing cloud computing services into RIBE is an interesting idea to alleviate the workloads of PKG and each user. The study of outsourcing RIBE (O-RIBE) was initiated by Li et al. [13], in which a semi-trusted key update cloud service provider is adopted to update each user's time key. Though Liang et al. [14] attempted to solve the same problems as in this work with proxy re-encryption technique, their scheme is insecure to resist the re-encryption key forgery attack and collusion attack [30]. To overcome the decryption challenges for users only with limited resources, Qin et al. [21] introduced a new revocation method called server-aided RIBE (SA-RIBE), contrary to previous O-RIBE, all workloads on users are now outscourced to the cloud server. Inspired by these two new models, Nguyen et al. [20] and Dong et al. [9] respectively designed the first lattice-based SA-RIBE scheme and the first lattice-based O-RIBE scheme. Recently, the generic constructions of RIBE with complete subset (CS) method and subset difference (SD) method were respectively proposed by Ma and Lin [16] and Lee [12].

Our Contributions. In this paper, we introduce two (and the first) constructions of lattice-based RIBE-CE. We inherit and extend the main efficiency and security advantages of Sun et al.'s model for RIBE: the system users do not have to store each time update key, as they can utilize the current short-term decryption key to decrypt all ciphertexts sent to him, meanwhile, the revoked users cannot access to both the previously and subsequently shared data in the cloud. Furthermore, PKG maintains a BT and adopts the CS method to handle key revocation with a logarithmic complexity workload in time key update, not growing linearly in the numbers of system users as in Sun et al. [27]. As for previous lattice-based RIBE schemes [7,20,29,11,28], our two constructions only work for one-bit message, but the multi-bit version can be achieved by adopting a standard transformation technique showed in [10,1]. As in [7,29,9,11,28], the public parameters almost enjoy the same asymptotic efficiency, though the bit-size of our final ciphertext in our two schemes is linear in the length of identity id, while all ciphertexts (the secure data) are stored in cloud, thus for the local users the storage cost is not a challenge. More startlingly, our two schemes are naturally SA-RIBE as [20] in which the recipient does not need to communicate with PKG for time update key, thus the recipient enjoys a lower decryption cost.

As in [20], each user's long-term private key of our second scheme is a trapdoor matrix, thus having a relatively large bit-size, but constant in the number of system users. In particular, the later scheme satisfies DKER, a default security requirement for RIBE, which is not considered by Sun et al.. A detailed comparison among the schemes [7,20,29,11,28] and ours is shown in Table 1.

As a high level, the design method of our first lattice-based RIBE-CE scheme is similar to the pairing-based concrete construction of Sun et al., and a double encryption mechanism is also adopted as the core building block. In our second scheme, instead of only using a conventional IBE scheme as in Sun et al., a lattice-based two-level hierarchical IBE (HIBE) scheme [1] is introduced, from which the PKG can issue a long-term private key (a trapdoor matrix, not a vector) to each user id. This technique enables a user id with a long-term private key to derive partial short-term decryption key corresponding to his identity id and time t all by himself, and thus achieving DKER.

Table 1. Comparison of Lattice-Based RIBE Schemes.

Schemes	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{sk}_{\mathsf{id}}\rvert$	$\lvert\mathsf{uk}_{\mathsf{t}}\rvert$	$\lvert\mathsf{dk}_{\mathsf{id},\mathsf{t}}\rvert$	$\lvert\mathsf{ct}_{\mathsf{id},\mathsf{t}}\rvert$	CE	DKER	Model
[7]	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n^2)$	$\mathcal{O}(r\log\frac{N}{r})\cdot\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	no	no	Standard
[20]	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n^2)$	$\mathcal{O}(r\log\frac{N}{r})\cdot\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	no	Unbounded	Standard
[29]	$\widetilde{\mathcal{O}}(n^2)$	$d\cdot\widetilde{\mathcal{O}}(n^2)$	$\mathcal{O}(r\log\frac{N}{r})\cdot\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	no	Bounded	Standard
[11]	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n^2)$	$\mathcal{O}(r\log\frac{N}{r})\cdot\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	no	Unbounded	Standard
[28]	$\widetilde{\mathcal{O}}(n^2)$	$d\cdot\widetilde{\mathcal{O}}(n^2)$	$\mathcal{O}(r\log\frac{N}{r})\cdot\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	no	Bounded	Quantum ROM
Ours-1	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$	$\mathcal{O}(r\log\frac{N}{r})\cdot\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n^2)$	yes	no	Standard
Ours-2	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n^2)$	$\mathcal{O}(r\log\frac{N}{r})\cdot\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n^2)$	yes	Unbounded	Standard

Note: n is a security parameter, $N = 2^n$ is the maximum numbers of system users, r is the number of revoked users, d is the number of private keys stored in each node over $\mathsf{path}(\mathsf{id})$; $|\cdot|$ denotes the bit-size, pp is public parameters, $\mathsf{sk}_{\mathsf{id}}$ is long-term private key, uk_{t} is time update key, $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$ is short-term decryption key, and $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$ is ciphertext; CE denotes ciphertext evolution and ROM denotes random oracle model.

Furthermore, looking into the details on time key update. The PKG in Sun et al. [27] issues a time update key for each non-revoked user with a conventional IBE scheme, and thus the workload of PKG grows linearly in the numbers of the non-revoked users. Instead, we adapt the classical BT-based revocation mechanism [19] to obtain a logarithmic complexity workload of PKG which serves as one solution to the challenge proposed by Sun et al., and an extended ciphertext design technique recently employed by Ma and Lin [16], which works as follows: the sender encrypts a message under a HIBE scheme and an IBE scheme corresponding to the time and node pairs (t, θ) where $\theta \in \mathsf{path}(\mathsf{id})$ and the recipient id is assigned to a leaf node in BT. Given a user revocation list (RL), PKG computes the time update key under the IBE scheme corresponding to the time and node pairs (t, θ') where $\theta' \in \mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$ denotes all the non-revoked children of revoked id with t, only a non-revoked id at t can derive a short-term decryption key consisting of two parts, which are corresponding to the two-level HIBE and the unique node $\theta \in \mathsf{path}(\mathsf{id}) \cap \mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$. Unfortunately, the bit-size of final ciphertext in our constructions is linear in the length of id.

The security of our lattice-based RIBE-CE schemes rely on a lattice-based IBE scheme [1] and a lattice-based double encryption scheme employed by Nguyen et al. [20] in the design of lattice-based SA-RIBE. The constructions are selectively secure in the standard model for our two schemes based on the hardness assumption of the learning with errors (LWE) problem.

Organization. The organization of the paper is as follows. In Section 2, we review the definition of RIBE-CE and some background knowledge on lattices. A lattice-based RIBE-CE scheme without DKER and a scheme with DKER in the standard model are described and analyzed in Sections 3 and 4, respectively. In the final Section 5, we conclude our whole paper.

2 Definition and Security Model

Table 2 refers to the notations used in this paper.

Table 2. Notations of This Paper.

Notation	Definition
$\mathbf{a}, \mathbf{A}$	Vectors, matrices
$\xleftarrow{\$}$	Sampling uniformly at random
$\|\cdot\|$, $\|\cdot\|_\infty$	Euclidean norm ℓ_2, infinity norm ℓ_∞
$\lceil e \rceil$, $\lfloor e \rceil$	The smallest integer not less than e, the integer closet to e
$\mathcal{O}$, $\widetilde{\mathcal{O}}$, ω	Standard asymptotic notations
$\log e$	Logarithm of e with base 2
ppt	Probabilistic polynomial-time

2.1 RIBE with Server-Aided Ciphertext Evolution

We first review the definition and security model of RIBE-CE introduced by Sun et al. [27]. An RIBE-CE is an extension of RIBE that supports key revocation, and additionally it delegates ciphertext evolution to a cloud server (Cloud). A trusted center first issues a master secret key (msk) and public parameters (pp). The PKG issues a long-term private key $\mathsf{sk}_{\mathsf{id}}$ for each system user id and a time update key uk_{t} with time t by using msk, meanwhile, it distributes uk_{t} to Cloud (and all users) and maintains a revocation list (RL) to record the state information on revoked users. The Cloud periodically transforms a ciphertext for id with t into a new one for $\mathsf{t}' > \mathsf{t}$ by using uk_{t}. To decrypt a ciphertext which specifies an identity id and a time t, the non-revoked recipient combines his long-term private key $\mathsf{sk}_{\mathsf{id}}$ and current time update key uk_{t} to derive a short-term decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$. The system model of RIBE-CE is shown in Fig. 1.

Definition 1. *An* RIBE-CE *scheme involves four distinct entities:* PKG*, a cloud server* Cloud*, sender and recipient, associated with identity space $\mathcal{I}$, time space $\mathcal{T}$ (time is treated as discrete and the size of $\mathcal{T}$ is polynomial in the security parameter) and message space $\mathcal{M}$, and consists of eight polynomial-time (*pt*) algorithms which are described as follows:*

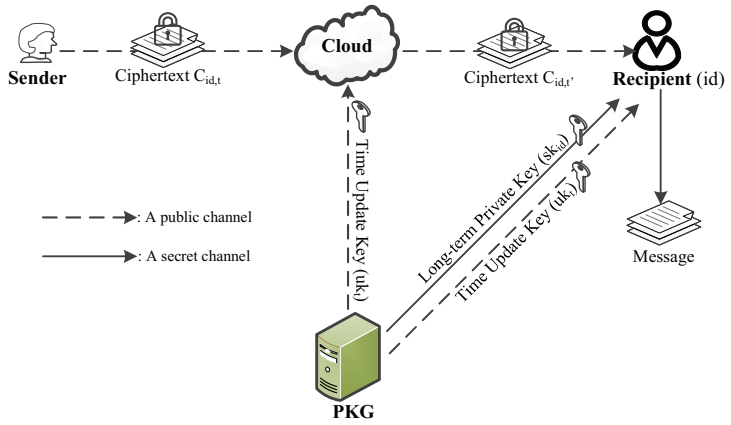

Fig. 1. System Model of RIBE-CE.

- $\mathsf{Setup}(1^n, N)$*: The setup algorithm is run by* PKG*, and it takes as input a security parameter* n *and the maximal number of system users* N*. It outputs a master secret key* msk*, the public parameters* pp*, a user revocation list* RL *(initially empty), and a state* st*. Note:* msk *is kept in secret by* PKG *and* pp *is made public and as an implicit input of all other algorithms.*
- $\mathsf{PriKeyGen}(\mathsf{msk}, \mathsf{id}, \mathsf{st})$*: The key generation algorithm is run by* PKG*, and it takes as input an identity* id*, the master secret key* msk*, and a state* st*. It outputs a long-term private key* $\mathsf{sk_{id}}$ *and an updated state* st*. Note:* $\mathsf{sk_{id}}$ *is sent to the recipient via a secret channel.*
- $\mathsf{KeyUpd}(\mathsf{RL}, \mathsf{t}, \mathsf{msk}, \mathsf{st})$*: The key update algorithm is run by* Cloud*, and it takes as input current revocation list* RL*, a time* t*, the master secret key* msk*, and a state* st*. It outputs a time update key* $\mathsf{uk_t}$*. Note:* $\mathsf{uk_t}$ *is sent to* Cloud *and all users via a public channel.*
- $\mathsf{DecKeyGen}(\mathsf{sk_{id}}, \mathsf{uk_t}, \mathsf{t})$*: The decryption key derivation algorithm is run by the recipient* id*, and it takes as input a long-term private key* $\mathsf{sk_{id}}$*, a corresponding time update key* $\mathsf{uk_t}$ *(or* $\perp$*), and the current time* t*. It outputs a short-term decryption key* $\mathsf{dk_{id,t}}$ *(or* $\perp$ *indicating that the recipient* id *was revoked).*
- $\mathsf{Encrypt}(\mathsf{id}, \mathsf{t}, \mathsf{m})$*: The encryption algorithm is run by the sender, and it takes as input a recipient's identity* id*, an encryption time* t*, and a message* m*. It outputs a ciphertext* $\mathsf{ct_{id,t}}$*.*
- $\mathsf{Evolve}(\mathsf{ct_{id,t}}, \mathsf{t'}, \mathsf{uk_t})$*: The ciphertext evolution algorithm is run by* Cloud*, and it takes as input a ciphertext* $\mathsf{ct_{id,t}}$ *with identity* id *and time* t*, a new time* $\mathsf{t'} > \mathsf{t}$*, and the current time update key* $\mathsf{uk_t}$*. It outputs a new ciphertext* $\mathsf{ct_{id,t'}}$*. Note: If* id *has been revoked at time* t*, the ciphertext remains unchanged.*
- $\mathsf{Decrypt}(\mathsf{dk_{id',t'}}, \mathsf{ct_{id,t}})$*: The decryption algorithm is run by the recipient, and it takes as input a ciphertext* $\mathsf{ct_{id,t}}$ *and a decryption key* $\mathsf{dk_{id',t'}}$*. It outputs a message* $\mathsf{m} \in \mathcal{M}$*, or a symbol* $\perp$*.*

– $\mathsf{Revoke}(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$*: The revocation algorithm is run by* PKG*, and it takes as input the current revocation list* RL*, an identity* id*, a revoked time* t*, and a state* st*. It outputs an updated revocation list* $\mathsf{RL} = \mathsf{RL} \cup \{(\mathsf{id}, \mathsf{t})\}$*. Note: a copy of* RL *will be sent to* Cloud *via a public channel.*

The correctness of an RIBE-CE scheme is described as follows: for all pp, msk, RL, and st generated by $\mathsf{Setup}(1^n, N)$, $\mathsf{sk_{id}}$ generated by $\mathsf{PriKeyGen}(\mathsf{msk}, \mathsf{id}, \mathsf{st})$ for $\mathsf{id} \in \mathcal{I}$, $\mathsf{uk_t}$ generated by $\mathsf{KeyUpd}(\mathsf{RL}, \mathsf{t}, \mathsf{msk}, \mathsf{st})$ for $\mathsf{t} \in \mathcal{T}$ and RL, $\mathsf{ct_{id,t}}$ generated by $\mathsf{Encrypt}(\mathsf{id}, \mathsf{t}, \mathsf{m})$ for $\mathsf{id} \in \mathcal{I}$, $\mathsf{t} \in \mathcal{T}$ and $\mathsf{m} \in \mathcal{M}$, and $\mathsf{ct_{id,t'}}$ generated by $\mathsf{Evolve}(\mathsf{ct_{id,t}}, \mathsf{t'}, \mathsf{uk_t})$, then it is required that:

– If $(\mathsf{id}, \mathsf{t'}) \notin \mathsf{RL}$ for all $\mathsf{t'} \leq \mathsf{t}$, then $\mathsf{DecKeyGen}(\mathsf{sk_{id}}, \mathsf{uk_t}, \mathsf{t}) = \mathsf{dk_{id,t}}$.
– If $(\mathsf{id}, \mathsf{t'}) \notin \mathsf{RL}$ for all $\mathsf{t'} < \mathsf{t}$, then $\mathsf{Evolve}(\mathsf{ct_{id,t'}}, \mathsf{t}, \mathsf{uk_t}) = \mathsf{ct_{id,t}}$.
– If $(\mathsf{id} = \mathsf{id'}) \wedge (\mathsf{t} = \mathsf{t'})$, then $\mathsf{Decrypt}(\mathsf{dk_{id',t'}}, \mathsf{ct_{id,t}}) = \mathsf{m}$.

Since RIBE-CE is an extension of RIBE, the indistinguishability under chosen-plaintext attack (ind-cpa) security of RIBE must be satisfied to guarantee message hiding security against an inside attacker $\mathcal{A}_0$ who owns a long-term private key (e.g., a revoked user), and an outside attacker $\mathcal{A}_1$ who knows all time update keys (e.g., the cloud server Cloud).

Sun et al. [27] defined the security against adaptive-revocable-identity-time chosen-plaintext attacks for RIBE-CE. Here, we only consider selective-revocable-identity-time security (a weaker notion initially was suggested in RIBE by Boldyreva et al. [4], subsequently by Chen et al. [7], Nguyen et al. [20] and Katsumata et al. [11], in which an adversary $\mathcal{A}$ ($\mathcal{A}_0$ or $\mathcal{A}_1$) sends a challenge identity and time pair $(\mathsf{id}^*, \mathsf{t}^*)$ to the challenger $\mathcal{C}$ before the execution of $\mathsf{Setup}(1^n, N)$. A slight difference is that we formalize the ind-cpa security adopting a game capturing a stronger privacy property called *indistinguishable from random* as defined in [1] and a stronger security property called DKER defined in [24].

In our ind-cpa security model of RIBE-CE, the attacker can request long-term private key, time update key, revocation, short-term decryption key (as in Sun et al., in our first construction, there is also no this query), and ciphertext evolution queries. One of the most restrictions of this model is that if the attacker has requested a long-term private key for the challenge identity id^*, then id^* must be revoked before (or at) the time update key query of challenge time t^*. Finally, the goal of the attacker is to determine that the challenge ciphertxet is completely random, or correctly encrypted on the challenge m^* corresponding to $(\mathsf{id}^*, \mathsf{t}^*)$. A detailed definition is described as follows:

Definition 2. *The* ind-cpa *security of* RIBE-CE *is shown in the following game:*

– Intial*: The adversary* $\mathcal{A}$ *first declares a challenge identity and time pair* $(\mathsf{id}^*, \mathsf{t}^*)$*.*
– Setup*: The challenger* $\mathcal{C}$ *runs* $\mathsf{Setup}(1^n, N)$ *to obtain* $(\mathsf{msk}, \mathsf{pp}, \mathsf{RL}, \mathsf{st})$*. Note:* RL *is initially empty,* $\mathcal{C}$ *keeps* msk *in secret by himself and provides* pp *to* $\mathcal{A}$*.*
– Query phase 1*: The query-answer between* $\mathcal{A}$ *and* $\mathcal{C}$ *is described in Table 3. Remark:* $\mathsf{PriKenGen}(\cdot), \mathsf{KeyUpd}(\cdot), \mathsf{Revoke}(\cdot), \mathsf{DecKeyGen}(\cdot)$ *and* CE *share* st *and the queries should be with some restrictions defined later. The* $\mathsf{DecKeyGen}(\cdot)$ *oracle is used to define* DKER *for our second scheme, which has not been provided by Sun et al..*

Table 3. The Query-Answer between $\mathcal{A}$ and $\mathcal{C}$.

	$\mathsf{PriKenGen}(\cdot)$	$\mathsf{KeyUpd}(\cdot)$	$\mathsf{Revoke}(\cdot)$	$\mathsf{DecKeyGen}(\cdot)$	CE
$\mathcal{A}$	id	RL, t	$\mathsf{RL}, \mathsf{id}, \mathsf{t}$	id, t	$\mathsf{ct}_{\mathsf{id},\mathsf{t}}, \mathsf{t}, \mathsf{t}'$
$\mathcal{C}$	$\mathsf{sk}_{\mathsf{id}}$	uk_{t}	$\mathsf{RL} = \mathsf{RL} \cup \{(\mathsf{id}, \mathsf{t})\}$	$\mathsf{dk}_{\mathsf{id},\mathsf{t}}$	$\mathsf{ct}_{\mathsf{id},\mathsf{t}'}$

- Challenge*: $\mathcal{A}$ submits a message $\mathsf{m}^* \in \mathcal{M}$. $\mathcal{C}$ samples a bit $b \xleftarrow{\$} \{0,1\}$. If $b = 0$, $\mathcal{C}$ returns a challenge ciphertext $\mathsf{ct}^*_{\mathsf{id}^*,\mathsf{t}^*}$ by running $\mathsf{Encrypt}(\mathsf{id}^*, \mathsf{t}^*, \mathsf{m}^*)$, otherwise, a random $\mathsf{ct}^*_{\mathsf{id}^*,\mathsf{t}^*} \xleftarrow{\$} \mathcal{U}$.*
- Query phase 2*: $\mathcal{A}$ can continue to make additional queries as before with the same restrictions.*
- Guess*: $\mathcal{A}$ outputs a bit $b^* \in \{0,1\}$, and wins if $b^* = b$.*

  In the above game, the following restrictions should be satisfied:
  - $\mathsf{KeyUpd}(\cdot)$ *and* $\mathsf{Revoke}(\cdot)$ *must be queried in a non-decreasing order of time.*
  - $\mathsf{Revoke}(\cdot)$ *cannot be queried at* t *if* $\mathsf{KeyUpd}(\cdot)$ *has been queried at* t.
  - $\mathsf{Revoke}(\cdot)$ *must be queried on* $(\mathsf{id}^*, \mathsf{t})$ *for* $\mathsf{t} \leq \mathsf{t}^*$ *if* $\mathsf{PriKenGen}(\cdot)$ *has been queried on* id^*.
  - $\mathsf{DecKeyGen}(\cdot)$ *cannot be queried at* t *if* $\mathsf{KeyUpd}(\cdot)$ *was not queried at* t.
  - $\mathsf{DecKeyGen}(\cdot)$ *cannot be queried on* $(\mathsf{id}^*, \mathsf{t}^*)$, *and in* CE *query,* $\mathsf{t}' > \mathsf{t}$.

$\mathcal{A}$'s advantage is defined as $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{RIBE\text{-}CE},\mathcal{A}}(n) = |\Pr[b^* = b] - 1/2|$. An RIBE-CE scheme is secure if $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{RIBE\text{-}CE},\mathcal{A}}(n)$ is negligible in the security parameter n.

2.2 Lattices

In this subsection, we recall the knowledge on integer lattices.

Definition 3. *Given n, m, $q \geq 2$, a random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{u} \in \mathbb{Z}_q^n$, the m-dimensional q-ary orthogonal lattice $\Lambda_q^{\perp}(\mathbf{A})$ (and its shift $\Lambda_q^{\mathbf{u}}(\mathbf{A})$) is defined as: $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{0} \bmod q\}$ and $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\}$.*

The discrete Gaussian distribution over Λ with the center $\mathbf{c} \in \mathbb{Z}^m$ and a Gaussian parameter $s > 0$ is denoted as $\mathcal{D}_{\Lambda,s,\mathbf{c}}$, and we omit the subscript and denote it as $\mathcal{D}_{\Lambda,s}$ if $\mathbf{c} = \mathbf{0}$.

Lemma 1 ([10]). *For integers n, $q \geq 2$, $m \geq 2n\lceil \log q \rceil$, assume that the columns of a random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generates $\mathbb{Z}_q^n$, let $\epsilon \in (0, 1/2)$ and $s \geq \eta_\epsilon(\Lambda^{\perp}(\mathbf{A}))$, then the followings hold:*

1. *For $\mathbf{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,s}$, the statistical distance between $\mathbf{u} = \mathbf{A} \cdot \mathbf{e} \bmod q$ and $\mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^n$ is at most 2ϵ.*
2. *For $\mathbf{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,s}$, then $\Pr[\|\mathbf{e}\|_\infty \leq \lceil s \cdot \log m \rceil]$ holds with a larger probability.*
3. *The min-entropy of $\mathcal{D}_{\mathbb{Z}^m,s}$ is at least $m - 1$.*

A ppt trapdoor generation algorithm returning a statistically close to uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a low Gram-Schmidt norm basis for $\Lambda_q^{\perp}(\mathbf{A})$ plays a key role in lattice-based cryptography. The algorithm was first introduced by Ajtai [2], and two improvements were investigated in [3,18].

Lemma 2 ([2,3,18]). *Let $n \geq 1$, $q \geq 2$, $m = 2n\lceil \log q \rceil$, there is a* ppt *algorithm* $\mathsf{TrapGen}(q, n, m)$ *that returns* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *statistically close to an uniform matrix in* $\mathbb{Z}_q^{n \times m}$ *and a trapdoor* $\mathbf{R_A}$ *for* $\Lambda_q^{\perp}(\mathbf{A})$.

Gentry et al. [10] showed an algorithm to sample short vectors (or matrices) from a discrete Gaussian distribution, and an improvement was given in [18]. Meanwhile, to delegate a trapdoor for a super-lattice was given in [6].

Lemma 3 ([10,18]). *Let $n \geq 1$, $q \geq 2$, $m = 2n\lceil \log q \rceil$, given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a trapdoor $\mathbf{R_A}$ for $\Lambda_q^{\perp}(\mathbf{A})$, a parameter $s = \omega(\sqrt{n \log q \log n})$, and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, there is a* ppt *algorithm* $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R_A}, \mathbf{u}, s)$ *returning a shorter vector $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}), s}$.*

Lemma 4 ([6]). *Let $q \geq 2$, $m = 2n\lceil \log q \rceil$, given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ who can generate $\mathbb{Z}_q^n$, a basis $\mathbf{R_A} \in \mathbb{Z}^{m \times m}$ for $\Lambda_q^{\perp}(\mathbf{A})$, a random $\mathbf{A}' \in \mathbb{Z}_q^{n \times m'}$, there is a deterministic algorithm* $\mathsf{ExtBasis}(\mathbf{R_A}, \mathbf{A}^* = \mathbf{A}|\mathbf{A}')$ *returning a basis $\mathbf{R_{A^*}} \in \mathbb{Z}^{(m+m') \times (m+m')}$ for $\Lambda_q^{\perp}(\mathbf{A}^*)$, especially, $\mathbf{R_A}$, $\mathbf{R_{A^*}}$ are with equal Gram-Schmidt norm. Note: the result holds for any given permutation of all columns of $\mathbf{A}^*$.*

Lemma 5 ([6]). *Let $n \geq 1$, $q \geq 2$, $m = 2n\lceil \log q \rceil$, $s \geq \|\widetilde{\mathbf{R_A}}\| \cdot \omega(\sqrt{\log n})$, $\mathbf{R_A} \in \mathbb{Z}^{m \times m}$ is a basis for $\Lambda_q^{\perp}(\mathbf{A})$, there is a* ppt *algorithm* $\mathsf{RandBasis}(\mathbf{R_A}, s)$ *returning a new basis $\mathbf{R'_A} \in \mathbb{Z}^{m \times m}$ and $\|\mathbf{R'_A}\| \leq s \cdot \sqrt{m}$. In particular, for two basis matrices $\mathbf{R_A^{(1)}}$ and $\mathbf{R_A^{(2)}}$ for $\Lambda_q^{\perp}(\mathbf{A})$, and $s \geq max\{\|\widetilde{\mathbf{R_A^{(1)}}}\|, \|\widetilde{\mathbf{R_A^{(2)}}}\|\} \cdot \omega(\sqrt{\log n})$,* $\mathsf{RandBasis}(\mathbf{R_A^{(1)}}, s)$ *is statistically close to* $\mathsf{RandBasis}(\mathbf{R_A^{(2)}}, s)$.

Lemma 6 ([1]). *Let $q > 2$, $m > n$, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{A}' \in \mathbb{Z}_q^{n \times m'}$, and $s > \|\widetilde{\mathbf{R_A}}\| \cdot \omega(\sqrt{\log(m+m')})$, given a trapdoor $\mathbf{R_A}$ for $\Lambda_q^{\perp}(\mathbf{A})$ and $\mathbf{u} \in \mathbb{Z}_q^n$, there is a* ppt *algorithm* $\mathsf{SampleLeft}(\mathbf{A}|\mathbf{A}', \mathbf{R_A}, \mathbf{u}, s)$ *returning a shorter $\mathbf{e} \in \mathbb{Z}^{2m}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}|\mathbf{A}'), s}$.*

Lemma 7 ([1]). *Let $q > 2$, $m > n$, $\mathbf{A}$, $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, $s > \|\widetilde{\mathbf{R_B}}\| \cdot \mathcal{O}(\sqrt{m}) \cdot \omega(\sqrt{\log m})$, given a trapdoor $\mathbf{R_B}$, a low-norm $\mathbf{R} \in \{-1, 1\}^{m \times m}$, and $\mathbf{u} \in \mathbb{Z}_q^n$, there is a* ppt *algorithm* $\mathsf{SampleRight}(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{R_B}, \mathbf{u}, s)$ *returning a shorter $\mathbf{e} \in \mathbb{Z}^{2m}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}), s}$, where $\mathbf{F} = [\mathbf{A}|\mathbf{AR} + \mathbf{B}]$.*

We recall the learning with errors (LWE) problem introduced by Regev [22].

Definition 4. *The* LWE *problem is defined as follows: given $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, a distribution χ over $\mathbb{Z}$, let $\mathcal{A}_{\mathbf{s},\chi}$ be the distribution $(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{e})$ where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{e} \xleftarrow{\$} \chi^m$, and to make distinguish between $\mathcal{A}_{\mathbf{s},\chi}$ and $\mathcal{U} \xleftarrow{\$} \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$. Let $\beta \geq \sqrt{n} \cdot \omega(\log n)$, for a prime power q, given a β-bounded χ, the* LWE *problem is as least as hard as the shortest independent vectors problem* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(nq/\beta)}$.

An injective encoding function $\mathcal{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ is adopted for our RIBE-CE schemes. An explicit design called encoding with full-rank differences (FRD) was proposed by Agrawal et al. [1].

Definition 5. *Let $n > 1$, prime $q \geq 2$, an injective encoding function $\mathcal{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ is called* FRD *if:*

1. *For $\forall \mathbf{e}_1, \mathbf{e}_2 \in \mathbb{Z}_q^n$, $\mathbf{e}_1 \neq \mathbf{e}_2$, $\mathcal{H}(\mathbf{e}_1) - \mathcal{H}(\mathbf{e}_2) \in \mathbb{Z}_q^{n \times n}$ is full-rank.*
2. *$\mathcal{H}$ can be computed in a polynomial time, i.e., $\mathcal{O}(n \log q)$.*

Two followings two facts will be used in the security proofs of this work.

Lemma 8 ([1]). *Let $n \geq 1$, prime $q > 2$, $m > (n+1)\log q + \omega(\log n)$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times k=poly(n)}$, and $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times k} \bmod q$. Then, for all $\mathbf{w} \in \mathbb{Z}_q^m$, $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top\mathbf{w})$ is statistically close to $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top\mathbf{w})$.*

Lemma 9 ([1]). *Let $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times m}$ and $\mathbf{w} \in \mathbb{R}^m$, $\Pr[\|\mathbf{R}\cdot\mathbf{w}\|_\infty > \|\mathbf{w}\|_\infty \cdot \sqrt{m}\cdot\omega(\sqrt{\log m})] < negl(m)$.*

3 Our Lattice-Based RIBE-CE Scheme without DKER

Our first RIBE-CE scheme adopts a lattice-based IBE scheme [1] from which the PKG issues a long-term private key to each user id and a time update key to Cloud for ciphertext evolution, a classical BT revocation mechanism [19] to alleviate the load of PKG (a logarithmic complexity and a user id is viewed as a leaf node of BT, each node in BT has an identifier which is a fixed and unique binary string*, and an extended ciphertext design method [16] to resolve the problem of the same state information of BT is used in PriKeyGen$(\cdot)$ and KeyUpd$(\cdot)$.

3.1 Description of the Scheme

As in Sun et al. [27], our lattice-based RIBE-CE scheme also consists of eight pt algorithms: Setup, PriKeyGen, KeyUpd, DecKeyGen, Encrypt, Evolve, Decrypt and Revoke. The main algorithms are described as follows:

- Setup$(1^n, N)$: On input a security parameter n and the maximal number of users $N = 2^n$, set prime modulus $q = \widetilde{\mathcal{O}}(n^3)$, dimension $m = 2nk$ where $k = \lceil \log q \rceil$, Gaussian parameter $s = \widetilde{\mathcal{O}}(m)$ and norm bound $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$ for a distribution χ. PKG specifies the following steps:
  1. Let identity space $\mathcal{I} = 0||\{0,1\}^n$, time space $\mathcal{T} \subset 0||\{0,1\}^n$, and message space $\mathcal{M} = \{0,1\}$.
  2. Run TrapGen(q, n, m) to generate $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a trapdoor $\mathbf{R_A}$, and $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ with a trapdoor $\mathbf{R_B}$.
  3. Sample a collision-resistance hash function $\mathcal{G} : \{0,1\}^* \to \mathbb{Z}_q^n$, and an FRD function $\mathcal{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$.
  4. Sample $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_0, \mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$.
  5. Set the sate st = BT that BT is with at least N leaf nodes, and the initial revocation list RL $= \emptyset$.

* Set an identifier of the root node (root) as 0, and an identifier of other node is assigned as follows: each edge in BT is assigned with 0 or 1 depending on whether it is connected to a left or right child node, thus an identifier of each node is defined as all labels of edges in the path from root to this node. Obviously, each user id $= (0, id_1, id_2, \cdots, id_{\log N}) \in 0||\{0,1\}^{\log N}$ is with a path path(id), where N is the maximal number of system users. Additionally, the detailed description of KUNodes(BT,RL,t) algorithm is omitted in this paper and any interested readers please refer to [4,7,24,21,20,9].

6. Set $\mathsf{pp} = (\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \mathbf{v}, \mathbf{U}, \mathcal{G}, \mathcal{H})$, and $\mathsf{msk} = (\mathbf{R}_\mathbf{A}, \mathbf{R}_\mathbf{B})$.
7. Output $(\mathsf{pp}, \mathsf{msk}, \mathsf{RL}, \mathsf{st})$ where msk is kept in *secret* by PKG, and pp is made *public* and as an implicit input of all other algorithms.

– $\mathsf{PriKeyGen}(\mathsf{msk}, \mathsf{id}, \mathsf{st})$: On input an identity $\mathsf{id} \in \mathcal{I}$, the master secret key msk and the state st. PKG specifies the following steps:
1. View id as an unassigned leaf node of BT, thus, $\mathsf{id} \in 0||\{0,1\}^n$.
2. Define $\mathbf{A}_\mathsf{id} = [\mathbf{A}|\mathbf{A}_0 + \mathcal{H}(\mathcal{G}(\mathsf{id}))\mathbf{A}_1] \in \mathbb{Z}_q^{n\times 2m}$.
3. Run $\mathsf{SampleLeft}(\mathbf{A}_\mathsf{id}, \mathbf{R}_\mathbf{A}, \mathbf{v}, s)$ to generate $\mathbf{e}_\mathsf{id} \in \mathbb{Z}^{2m}$ satisfying $\mathbf{A}_\mathsf{id} \cdot \mathbf{e}_\mathsf{id} = \mathbf{v} \bmod q$.
4. Output an updated state st, and $\mathsf{sk}_\mathsf{id} = \mathbf{e}_\mathsf{id}$. *Note*: sk_id is sent to user id via a *secret* channel.

– $\mathsf{KeyUpd}(\mathsf{RL}, \mathsf{t}, \mathsf{msk}, \mathsf{st})$: On input a time $\mathsf{t} \in \mathcal{T}$, the master secret key msk, a revocation list RL and the state st. PKG specifies the following steps:
1. For $\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t})$, define $\mathbf{B}_{\mathsf{t}_\theta} = [\mathbf{B}|\mathbf{B}_0 + \mathcal{H}(\mathcal{G}(\mathsf{t}||\theta))\mathbf{B}_1] \in \mathbb{Z}_q^{n\times 2m}$.
2. Run $\mathsf{SampleLeft}(\mathbf{B}_{\mathsf{t}_\theta}, \mathbf{R}_\mathbf{B}, \mathbf{U}, s)$ to generate $\mathbf{E}_\theta \in \mathbb{Z}^{2m\times k}$ satisfying $\mathbf{B}_{\mathsf{t}_\theta} \cdot \mathbf{E}_\theta = \mathbf{U} \bmod q$.
3. Output $\mathsf{uk}_\mathsf{t} = (\theta, \mathbf{E}_\theta)_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$.

– $\mathsf{DecKeyGen}(\mathsf{sk}_\mathsf{id}, \mathsf{uk}_\mathsf{t}, \mathsf{t})$: On input a long-term private key $\mathsf{sk}_\mathsf{id} = \mathbf{e}_\mathsf{id}$, a time t and current time update key $\mathsf{uk}_\mathsf{t} = (\theta, \mathbf{E}_\theta)_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$. The recipient id specifies the following steps:
1. If $\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}) = \emptyset$, return $\perp$ and abort.
2. Otherwise, select $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$ (only one θ exists).
3. Return $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = (\mathbf{e}_\mathsf{id}, \mathbf{E}_\theta)$.

– $\mathsf{Encrypt}(\mathsf{id}, \mathsf{t}, \mathsf{m})$: On input an identity $\mathsf{id} \in \mathcal{I}$, a time $\mathsf{t} \in \mathcal{T}$, and a message $\mathsf{m} \in \{0,1\}$. The sender will specify the following steps:
1. Let $\mathbf{A}_\mathsf{id} = [\mathbf{A}|\mathbf{A}_0 + \mathcal{H}(\mathcal{G}(\mathsf{id}))\mathbf{A}_1] \in \mathbb{Z}_q^{n\times 2m}$.
2. For $\theta \in \mathsf{path}(\mathsf{id})$, define $\mathbf{B}_{\mathsf{id}_\theta,\mathsf{t}} = [\mathbf{B}|\mathbf{B}_0 + \mathcal{H}(\mathcal{G}(\mathsf{t}||\theta))\mathbf{B}_1] \in \mathbb{Z}_q^{n\times 2m}$.
3. Sample $\mathbf{s}_0, \mathbf{s}_0' \xleftarrow{\$} \mathbb{Z}_q^n$, $e_0 \xleftarrow{\$} \chi$, $\mathbf{e}_0' \xleftarrow{\$} \chi^k$, $\mathbf{e}_1, \mathbf{e}_1' \xleftarrow{\$} \chi^m$, and $\mathbf{R}_1, \mathbf{R}_2 \xleftarrow{\$} \{1,-1\}^{m\times m}$.
4. Let $c_0 = \mathbf{v}^\mathrm{T}\mathbf{s}_0 + e_0 + \mathsf{m}\lfloor \frac{q}{2} \rfloor \bmod q \in \mathbb{Z}_q$, $\mathbf{c}_1 = \mathbf{A}_\mathsf{id}^\mathrm{T}\mathbf{s}_0 + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{R}_1^\mathrm{T}\mathbf{e}_1 \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
5. Let $\mathbf{c}_0' = \mathbf{U}^\mathrm{T}\mathbf{s}_0' + \mathbf{e}_0' + \mathsf{bin}(c_0)\lfloor \frac{q}{2} \rfloor \bmod q \in \mathbb{Z}_q^k$, $\mathbf{c}_{2,\theta} = \mathbf{B}_{\mathsf{id}_\theta,\mathsf{t}}^\mathrm{T}\mathbf{s}_0' + \begin{bmatrix} \mathbf{e}_1' \\ \mathbf{R}_2^\mathrm{T}\mathbf{e}_1' \end{bmatrix} \in \mathbb{Z}_q^{2m}$. *Note*: a binary decomposition function $\mathsf{bin} : \mathbb{Z}_q \to \{0,1\}^k$ is adopted here, and for all $\mathbf{e} \in \mathbb{Z}_q$ we have that $\mathbf{e} = (1, 2, \cdots, 2^{k-1}) \cdot \mathsf{bin}(\mathbf{e})$.
6. Output $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})}) \in (0||\{0,1\}^n)^2 \times \mathbb{Z}_q^k \times (\mathbb{Z}_q^{2m})^{n+2}$.

– $\mathsf{Evolve}(\mathsf{ct}_{\mathsf{id},\mathsf{t}}, \mathsf{t}', \mathsf{uk}_\mathsf{t})$: On input a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})})$, a new $\mathsf{t}' > \mathsf{t}$, and the current time update key $\mathsf{uk}_\mathsf{t} = (\theta, \mathbf{E}_\theta)_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$. The Cloud specifies the following steps:
1. If the recipient of $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$ has been revoked before (or at) time t, set $\mathsf{ct}_{\mathsf{id},\mathsf{t}'} = \mathsf{ct}_{\mathsf{id},\mathsf{t}}$.

2. Otherwise, compute $\mathbf{w}_0 = \mathbf{c}'_0 - \mathbf{E}_\theta^{\mathrm{T}} \cdot \mathbf{c}_{2,\theta} \bmod q \in \mathbb{Z}_q^k$, here, $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
3. Compute $c_0 = (1, 2, \cdots, 2^{k-1}) \cdot \lfloor \frac{2}{q}\mathbf{w}_0 \rfloor \in \mathbb{Z}_q$.
4. Sample $\mathbf{s}''_0 \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}''_0 \xleftarrow{\$} \chi^k$, $\mathbf{e}''_1 \xleftarrow{\$} \chi^m$, and $\mathbf{R}'_2 \xleftarrow{\$} \{1, -1\}^{m \times m}$.
5. For $\theta \in \mathsf{path}(\mathsf{id})$, define $\mathbf{B}_{\mathsf{id}_\theta, \mathsf{t}'} = [\mathbf{B} | \mathbf{B}_0 + \mathcal{H}(\mathcal{G}(\mathsf{t}' || \theta))\mathbf{B}_1] \in \mathbb{Z}_q^{n \times 2m}$.
6. Let $\mathbf{c}''_0 = \mathbf{U}^{\mathrm{T}}\mathbf{s}''_0 + \mathbf{e}''_0 + \mathsf{bin}(c_0)\lfloor \frac{q}{2} \rfloor \bmod q \in \mathbb{Z}_q^k$, and $\mathbf{c}'_{2,\theta} = \mathbf{B}_{\mathsf{id}_\theta, \mathsf{t}'}^{\mathrm{T}}\mathbf{s}''_0 + \begin{bmatrix} \mathbf{e}''_1 \\ \mathbf{R}_2'^{\mathrm{T}}\mathbf{e}''_1 \end{bmatrix} \bmod q \in \mathbb{Z}_q^{2m}$.
7. Output $\mathsf{ct}_{\mathsf{id},\mathsf{t}'} = (\mathsf{id}, \mathsf{t}', \mathbf{c}''_0, \mathbf{c}_1, (\mathbf{c}'_{2,\theta})_{\theta \in \mathsf{path}(\mathsf{id})}) \in (0 || \{0,1\}^n)^2 \times \mathbb{Z}_q^k \times (\mathbb{Z}_q^{2m})^{n+2}$.

– $\mathsf{Decrypt}(\mathsf{dk}_{\mathsf{id}',\mathsf{t}'}, \mathsf{ct}_{\mathsf{id},\mathsf{t}})$: On input a ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}'_0, \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta \in \mathsf{path}(\mathsf{id})})$ and a decryption key $\mathsf{dk}_{\mathsf{id}',\mathsf{t}'}$. The recipient id' specifies the following steps:
1. If $(\mathsf{id} \neq \mathsf{id}') \vee (\mathsf{t} \neq \mathsf{t}')$, return $\perp$ and abort.
2. Otherwise, compute $\mathbf{w}_0 = \mathbf{c}'_0 - \mathbf{E}_\theta^{\mathrm{T}} \cdot \mathbf{c}_{2,\theta} \bmod q \in \mathbb{Z}_q^k$, here, $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
3. Compute $c_0 = (1, 2, \cdots, 2^{k-1}) \cdot \lfloor \frac{2}{q}\mathbf{w}_0 \rceil \in \mathbb{Z}_q$, and $w = c_0 - \mathbf{e}_{\mathsf{id}}^{\mathrm{T}}\mathbf{c}_1 \in \mathbb{Z}_q$.
4. Output $\lfloor \frac{2}{q} w \rceil \in \{0, 1\}$.

– $\mathsf{Revoke}(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$: On input current revocation list RL, an identity id, a time t, and a state $\mathsf{st} = \mathsf{BT}$. PKG specifies the following steps:
1. Add $(\mathsf{id}, \mathsf{t})$ to RL for all nodes associated with id.
2. Output an updated $\mathsf{RL} = \mathsf{RL} \cup \{(\mathsf{id}, \mathsf{t})\}$.

3.2 Analysis

We analysis the efficiency, correctness and security of our lattice-based RIBE-CE scheme without DKER.

Efficiency: The efficiency aspect of our lattice-based RIBE-CE without DKER with $N = 2^n$ is as follows:
– The bit-size of public parameters pp is $(6nm + n + nk + 2n) \log q = \widetilde{\mathcal{O}}(n^2)$.
– The long-term private key $\mathsf{sk}_{\mathsf{id}}$ has a short vector of bit-size $\widetilde{\mathcal{O}}(n)$.
– The time update key uk_{t} has bit-size $\mathcal{O}(r \log \frac{N}{r}) \cdot \widetilde{\mathcal{O}}(n)$ where r is the number of revoked users.
– The ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$ has bit-size $2(n+1) + (k + 2m(n+2)) \log q = \widetilde{\mathcal{O}}(n^2)$.
– The short-term decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$ has bit-size $\widetilde{\mathcal{O}}(n)$.

Correctness: If the first lattice-based RIBE-CE scheme is operated correctly as specified, and a recipient id is not revoked at time $\mathsf{t} \in \mathcal{T}$, then $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = (\mathbf{e}_{\mathsf{id}}, \mathbf{E}_\theta)$ satisfies $\mathbf{A}_{\mathsf{id}} \cdot \mathbf{e}_{\mathsf{id}} = \mathbf{v} \bmod q$ and $\mathbf{B}_{\mathsf{t}_\theta} \cdot \mathbf{E}_\theta = \mathbf{U} \bmod q$. In the decryption algorithm, the non-revoked id tries to derive m by using $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$:

Given a ciphertext (no matter an original or evolutive ciphertext) $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}'_0, \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta \in \mathsf{path}(\mathsf{id})})$.

1. Parse $\mathbf{c}_{2,\theta} = \begin{bmatrix} \mathbf{c}_{2,0} \\ \mathbf{c}_{2,1} \end{bmatrix}$ where $\mathbf{c}_{2,i \in \{0,1\}} \in \mathbb{Z}_q^m$, $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
2. Compute

$$\mathbf{w}_0 = \mathbf{c}'_0 - \mathbf{E}_\theta^{\mathrm{T}}\mathbf{c}_{2,\theta} = \mathbf{U}^{\mathrm{T}}\mathbf{s}'_0 + \mathbf{e}'_0 + \mathsf{bin}(c_0)\lfloor\frac{q}{2}\rfloor - \mathbf{E}_\theta^{\mathrm{T}}\begin{bmatrix}\mathbf{c}_{2,0}\\ \mathbf{c}_{2,1}\end{bmatrix}$$
$$= \mathsf{bin}(c_0)\lfloor\frac{q}{2}\rfloor + \underbrace{\mathbf{e}'_0 - \mathbf{E}_\theta^{\mathrm{T}}\begin{bmatrix}\mathbf{e}'_1\\ \mathbf{R}_2^{\mathrm{T}}\mathbf{e}'_1\end{bmatrix}}_{\mathsf{error}'}$$

According to our parameters settings, it can be checked that the error term error' is bounded by $q/5$ (i.e., $\|\mathsf{error}'\|_\infty < q/5$), thus, we have the conclusion $(1, 2, \cdots, 2^{k-1}) \cdot \lfloor\frac{2}{q}\mathbf{w}_0\rceil = c_0$ with overwhelming probability.

3. Parse $\mathbf{c}_1 = \begin{bmatrix}\mathbf{c}_{1,0}\\ \mathbf{c}_{1,1}\end{bmatrix}$ where $\mathbf{c}_{1,i\in\{0,1\}} \in \mathbb{Z}_q^m$, and compute

$$w = c_0 - \mathbf{e}_{\mathsf{id}}^{\mathrm{T}}\mathbf{c}_1 = \mathbf{v}^{\mathrm{T}}\mathbf{s}_0 + e_0 + \mathsf{m}\lfloor\frac{q}{2}\rfloor - \mathbf{e}_{\mathsf{id}}^{\mathrm{T}}\begin{bmatrix}\mathbf{c}_{1,0}\\ \mathbf{c}_{1,1}\end{bmatrix} = \mathsf{m}\lfloor\frac{q}{2}\rfloor + \underbrace{e_0 - \mathbf{e}_{\mathsf{id}}^{\mathrm{T}}\begin{bmatrix}\mathbf{e}_1\\ \mathbf{R}_1^{\mathrm{T}}\mathbf{e}_1\end{bmatrix}}_{\mathsf{error}}$$

According to our parameters settings, it can be checked that the error term error is bounded by $q/5$ (i.e., $\|\mathsf{error}\|_\infty < q/5$), thus, we have the conclusion $\lfloor\frac{2}{q}w\rceil = \mathsf{m}$ with overwhelming probability.

Theorem 1. *Our lattice-based* RIBE-CE *without* DKER *is* ind-cpa *secure if the* LWE *assumption holds.*

Proof. To proof this theorem, we define a list of games where the first one is identical to the original ind-cpa game as in Definition 2 and show that a ppt adversary $\mathcal{A}$ has advantage zero in the last game. We show that $\mathcal{A}$ cannot distinguish between these games, and thus $\mathcal{A}$ has negligible advantage in winning the original ind-cpa game. In particular, the LWE hardness assumption is adopted to prove that Game 2 and Game 3 are indistinguishable.

Let id^* be a challenge identity and t^* be a challenge time, we consider two types of adversaries:

- Type-0: An inside adversary $\mathcal{A}_0$ who requests a long-term private key on the challenge identity id^*. In this case, id^* must be revoked at $\mathsf{t} \le \mathsf{t}^*$.
- Type-1: An outside adversary $\mathcal{A}_1$ who only requests a long-term private key on the identity $\mathsf{id} \neq \mathsf{id}^*$.

For Type-0 adversary, we simulate the game as follow:

Game 0. It is the original ind-cpa game defined in Definition 2.

Game 1. We slightly change the way that $\mathcal{C}_0$ generates $\mathbf{B}_0$ in pp. $\mathcal{C}_0$ samples $\mathbf{R}_2^* \xleftarrow{\$} \{1, -1\}^{m\times m}$ mod q at the setup phase, and defines $\mathbf{B}_0 = \mathbf{B}\mathbf{R}_2^* - \mathcal{H}(\mathcal{G}(\mathsf{t}^*||\theta))\mathbf{B}_1$ mod q. For the remainders, they are unchanged and identical to those in Game 0. Next, we show that Game 0 and Game 1 are indistinguishable. In Game 1, $\mathbf{R}_2^*$ is used only in the designs of $\mathbf{B}_0$, $\mathbf{R}_2^{*\mathrm{T}}\mathbf{e}'_1$. According to Lemma 8, $(\mathbf{B}, \mathbf{B}\mathbf{R}_2^*, \mathbf{R}_2^{*\mathrm{T}}\mathbf{e}'_1)$ is statistically close to $(\mathbf{B}, \mathbf{C}_0, \mathbf{R}_2^{*\mathrm{T}}\mathbf{e}'_1)$, where $\mathbf{C}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. In $\mathcal{A}_0$'s view, $\mathbf{B}\mathbf{R}_2^*$ is statistically close to uniform, and thus $\mathbf{B}_0$ is close to uniform. Hence, $\mathbf{B}_0$ in Game 1 and Game 0 are indistinguishable.

Game 2: We redesign $\mathbf{B}$ and $\mathbf{B}_1$. $\mathcal{C}_0$ samples $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and runs $\mathsf{TrapGen}(q, n, m)$ to obtain $\mathbf{B}_1$ with a trapdoor $\mathbf{R}_{\mathbf{B}_1}$. Let $\mathbf{B}_{\mathsf{t}_\theta} = [\mathbf{B}|\mathbf{B}\mathbf{R}_2^* + (\mathcal{H}(\mathcal{G}(\mathsf{t}||\theta)) - \mathcal{H}(\mathcal{G}(\mathsf{t}^*||\theta^*)))\mathbf{B}_1]$, due to the collision-resistance property of $\mathcal{G}$ and the main

property of FRD, $\mathcal{H}(\mathcal{G}(\mathsf{t}||\theta)) - \mathcal{H}(\mathcal{G}(\mathsf{t}^*||\theta^*))$ is full-rank, and $\mathbf{R}_{\mathbf{B}_1}$ is also a trapdoor for $\Lambda_q^{\perp}((\mathcal{H}(\mathcal{G}(\mathsf{t}||\theta)) - \mathcal{H}(\mathcal{G}(\mathsf{t}^*||\theta^*)))\mathbf{B}_1)$. $\mathcal{C}_0$ responds a time update key query for any $\mathsf{t} \neq \mathsf{t}^*$ (id^* has been revoked before or at t^*) by running $\mathsf{SampleRight}(\mathbf{B}, (\mathcal{H}(\mathcal{G}(\mathsf{t}||\theta)) - \mathcal{H}(\mathcal{G}(\mathsf{t}^*||\theta^*)))\mathbf{B}_1, \mathbf{R}_2^*, \mathbf{R}_{\mathbf{B}_1}, \mathbf{U}, s)$ that returns $\mathbf{E}_{\theta \in \mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$. Additionally, the parameter $s = \widetilde{\mathcal{O}}(m)$ is sufficiently large. According to Lemma 7, $\mathbf{E}_\theta$ is statistically close to that in Game 1. For the remainders, they are unchanged and identical to those in Game 1. Because $\mathbf{B}$ and $\mathbf{B}_1$ are statistically close to those in Game 1, the advantage of $\mathcal{A}_0$ in Game 2 is at most negligibly different from that in Game 1.

Game 3: We redesign the partial challenge ciphertexts $\mathbf{c}_0'^*$ and $\mathbf{c}_{2,\theta}^*$, and the remainders (including $\mathbf{c}_1^*$) are unchanged and identical to those in Game 2. $\mathcal{C}_0$ samples $\mathbf{c}_0'^* \xleftarrow{\$} \mathbb{Z}_q^k$ and $\mathbf{c}_{2,\theta}^* \xleftarrow{\$} \mathbb{Z}_q^{2m}$. Because $\mathbf{c}_0'^*$ and $\mathbf{c}_{2,\theta}^*$ are always random, the advantage of $\mathcal{A}_0$ in returning a correct c_0 is zero, and the same advantage zero for the message $\mathsf{m} = \lfloor \frac{2}{q}(c_0 - \mathbf{e}_{\mathsf{id}^*}^{\mathrm{T}}\mathbf{c}_1^*) \rceil$ is returned correctly.

A reduction from the LWE problem will be given to show that Game 2 and Game 3 are computationally indistinguishable for a ppt adversary.

A reduction from LWE: Assume that there is a ppt adversary $\mathcal{A}_0$ distinguishing Game 2 and Game 3 with non-negligible advantage, then we use $\mathcal{A}_0$ to design an algorithm $\mathcal{B}_0$ to solve the LWE problem defined in Definition 4.

Given an LWE instance, a fresh pair $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for $i = 1, \cdots, m(n+1)+k$, from a sampling oracle, which is truly random $\mathcal{R}_\$$ or noisy pseudo-random $\mathcal{R}_{\mathbf{s}_0'}$ for a secret vector $\mathbf{s}_0' \in \mathbb{Z}_q^n$. The target of $\mathcal{B}_0$ is to distinguish between the two oracles by utilizing $\mathcal{A}_0$ as follows:

Instance: $\mathcal{B}_0$ receives an LWE instance (i.e., $(\mathbf{a}_i, b_i), i = 1, \cdots, m(n+1)+k$).

Setup: $\mathcal{B}_0$ does as follows:

1. Assemble $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ from m of the given LWE samples, that is, define $\mathbf{B} = [\mathbf{a}_1|\mathbf{a}_2|\cdots|\mathbf{a}_m]$.
2. Assemble $\mathbf{U} \in \mathbb{Z}_q^{n \times k}$ from the unused samples, define $\mathbf{U} = [\mathbf{a}_{m+1}|\cdots|\mathbf{a}_{m+k}]$.
3. Run $\mathsf{TrapGen}(q, n, m)$ to generate $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor $\mathbf{R}_{\mathbf{A}}$, sample $\mathbf{A}_0, \mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$.
4. Design the remainders of public matrices, $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{Z}_q^{n \times m}$, as in Game 2 by using id^*, t^*, and R_2^*.
5. Sample a collision-resistance hash function $\mathcal{G} : \{0,1\}^* \to \mathbb{Z}_q^n$, and an FRD function $\mathcal{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$.
6. Let $\mathsf{pp} = (\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{B}_0, \mathbf{v}, \mathbf{B}_1, \mathbf{U}, \mathcal{G}, \mathcal{H})$, and send pp to $\mathcal{A}_1$.

Queries: $\mathcal{B}_0$ answers a time update key query for t as in Game 2. As $\mathcal{B}_0$ knows the master secret key $\mathbf{R}_{\mathbf{A}}$, it can answer a long-term private key query for id (a shorter vector $\mathbf{e}_{\mathsf{id}}$) as in the real game.

Challenge: Once receive a message $\mathsf{m}^* \in \mathcal{M}$, $\mathcal{B}_0$ computes a challenge ciphertext for id^* and t^* as follows:

1. Let $\mathbf{A}_{\mathsf{id}^*} = [\mathbf{A}|\mathbf{A}_0 + \mathcal{H}(\mathcal{G}(\mathsf{id}^*))\mathbf{A}_1] \in \mathbb{Z}_q^{n \times 2m}$.
2. Sample $\mathbf{s}_0 \xleftarrow{\$} \mathbb{Z}_q^n$, $e_0 \xleftarrow{\$} \chi$, $\mathbf{e}_1 \xleftarrow{\$} \chi^m$, and $\mathbf{R}_1 \xleftarrow{\$} \{1,-1\}^{m \times m}$.
3. Let $c_0^* = \mathbf{v}^{\mathrm{T}}\mathbf{s}_0 + e_0 + \mathsf{m}^* \lfloor \frac{q}{2} \rfloor \bmod q \in \mathbb{Z}_q$, $\mathbf{c}_1^* = \mathbf{A}_{\mathsf{id}^*}^{\mathrm{T}}\mathbf{s}_0 + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{R}_1^{\mathrm{T}}\mathbf{e}_1 \end{bmatrix} \in \mathbb{Z}_q^{2m}$.

4. Assemble $\mathbf{e}'^*_{1,\theta}$ from m of the given LWE samples, define $\mathbf{e}'^*_{1,\theta} = \begin{bmatrix} b_{|\theta|m+1} \\ \vdots \\ b_{|\theta|m+m} \end{bmatrix}$, $\mathbf{b}^* = \begin{bmatrix} b_{m(n+1)+1} \\ \vdots \\ b_{m(n+1)+k} \end{bmatrix}$, where $\theta \in \mathsf{path}(\mathsf{id}^*)$ and $|\theta|$ denotes the length of θ.
5. Blind the message string by defining $\mathbf{c}'^*_0 = \mathbf{b}^* + \mathsf{bin}(c^*_0)\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^k$, and $\mathbf{c}^*_{2,\theta} = \begin{bmatrix} \mathbf{e}'^*_{1,\theta} \\ \mathbf{R}_2^{*\mathrm{T}}\mathbf{e}'^*_{1,\theta} \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
6. Send $\mathsf{ct}_{\mathsf{id}^*,\mathsf{t}^*} = (\mathsf{id}^*, \mathsf{t}^*, \mathbf{c}'^*_0, \mathbf{c}^*_1, (\mathbf{c}^*_{2,\theta})_{\theta \in \mathsf{path}(\mathsf{id}^*)})$ to $\mathcal{A}_0$.

We first show that if the LWE instance is from a noisy pseudo-random $\mathcal{R}_{\mathbf{s}'_0}$, so $(\mathbf{c}^*_0, \mathbf{c}^*_1) \in \mathsf{ct}_{\mathsf{id}^*,\mathsf{t}^*}$ enjoys a distribution exactly as in Game 2. First, it can be checked that $\mathbf{B}_{\mathsf{id}^*_\theta,\mathsf{t}^*} = [\mathbf{B}|\mathbf{B}\mathbf{R}^*_2]$. Second, it can be checked that $\mathbf{e}'^*_{1,\theta} = \mathbf{B}^{\mathrm{T}}\mathbf{s}'_0 + \mathbf{e}'_1 \bmod q$ where $\mathbf{e}'_1 \xleftarrow{\$} \chi^m$. Thus, $\mathbf{c}^*_{2,\theta}$ enjoys the following structure:

$$\mathbf{c}^*_{2,\theta} = \begin{bmatrix} \mathbf{e}'^*_{1,\theta} \\ \mathbf{R}_2^{*\mathrm{T}}\mathbf{e}'^*_{1,\theta} \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{\mathrm{T}}\mathbf{s}'_0 + \mathbf{e}'_1 \\ (\mathbf{B}\mathbf{R}^*_2)^{\mathrm{T}}\mathbf{s}'_0 + \mathbf{R}_2^{*\mathrm{T}}\mathbf{e}'_1 \end{bmatrix} = \mathbf{B}^{\mathrm{T}}_{\mathsf{id}^*_\theta,\mathsf{t}^*}\mathbf{s}'_0 + \begin{bmatrix} \mathbf{e}'_1 \\ \mathbf{R}_2^{*\mathrm{T}}\mathbf{e}'_1 \end{bmatrix} \bmod q,$$

which implies that $\mathbf{c}^*_{2,\theta}$ is exactly the $\mathbf{c}_{2,\theta}$ part of a valid challenge ciphertext in Game 2.

We then show that if the LWE instance is from a truly random $\mathcal{R}_\$$, then $(\mathbf{c}'^*_0, (\mathbf{c}^*_{2,\theta})_{\theta \in \mathsf{path}(\mathsf{id}^*)}) \in \mathsf{ct}_{\mathsf{id}^*,\mathsf{t}^*}$ enjoys a distribution exactly as in Game 3. It can be checked that $\mathbf{b}^*$ is unform over $\mathbb{Z}_q^k$, and $\mathbf{e}'^*_{1,\theta}$ are unform over $\mathbb{Z}_q^m$. Thus, $\mathbf{c}^*_{2,\theta}$ is unform and independent over $\mathbb{Z}_q^{2m}$, which implies that $\mathbf{c}^*_{2,\theta}$ is exactly the $\mathbf{c}_{2,\theta}$ part of a valid challenge ciphertext in Game 3.

Guess: After making some additional queries, $\mathcal{A}_0$ returns a guess for which challenger, Game 2 or Game 3, it is interacting with. Then, $\mathcal{B}_0$ returns the guess of $\mathcal{A}_0$ as an answer to the given LWE instance.

According to the above analysis, if the LWE instance is from $\mathcal{R}_{\mathbf{s}'_0}$, $\mathcal{A}_0$'s view is as in Game 2, and if the LWE instance is from $\mathcal{R}_\$$, $\mathcal{A}_0$'s view is as in Game 3, and thus, the advantage of $\mathcal{B}_0$ in solving the LWE problem is the same as that of $\mathcal{A}_0$ in distinguishing Game 2 and Game 3.

For $\mathsf{Type\text{-}1}$ adversary, we simulate the game as follow:

Game 0. It is the original $\mathsf{ind\text{-}cpa}$ game defined in Definition 2.

Game 1. We slightly change the way that $\mathcal{C}_1$ generates $\mathbf{A}_0$ in pp. $\mathcal{C}_1$ samples $\mathbf{R}^*_1 \xleftarrow{\$} \{1, -1\}^{m \times m}$ at the setup phase, let $\mathbf{A}_0 = \mathbf{A}\mathbf{R}^*_1 - \mathcal{H}(\mathcal{G}(\mathsf{id}^*))\mathbf{A}_1 \bmod q$. For the remainders, they are unchanged and identical to those in Game 0. Next, we show that Game 0 and Game 1 are indistinguishable. In Game 1, $\mathbf{R}^*_1$ is used only in the designs of $\mathbf{A}_0$ and $\mathbf{R}_1^{*\mathrm{T}}\mathbf{e}_1$. So, according to Lemma 10, $(\mathbf{A}, \mathbf{A}\mathbf{R}^*_1, \mathbf{R}_1^{*\mathrm{T}}\mathbf{e}_1)$ is statistically close to $(\mathbf{A}, \mathbf{C}_1, \mathbf{R}_1^{*\mathrm{T}}\mathbf{e}_1)$, where $\mathbf{C}_1 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$. In $\mathcal{A}_1$'s view, $\mathbf{A}\mathbf{R}^*_1$ is statistically close to uniform, and thus $\mathbf{A}_0$ is close to uniform. Hence, $\mathbf{A}_0$ in Game 1 and Game 0 are indistinguishable.

Game 2: We redesign $\mathbf{A}$, $\mathbf{A}_1$. $\mathcal{C}_2$ samples $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and runs $\mathsf{TrapGen}(q,n,m)$ to get $\mathbf{A}_1$ with trapdoor $\mathbf{R}_{\mathbf{A}_1}$. Let $\mathbf{A}_{\mathsf{id}} = [\mathbf{A}|\mathbf{A}\mathbf{R}_0^* + (\mathcal{H}(\mathcal{G}(\mathsf{id})) - \mathcal{H}(\mathcal{G}(\mathsf{id}^*)))\mathbf{A}_1]$, and due to the collision-resistance property of $\mathcal{G}$ and the main property of FRD, $\mathcal{H}(\mathcal{G}(\mathsf{id}) - \mathcal{H}(\mathcal{G}(\mathsf{id}^*))$ is full-rank, $\mathbf{R}_{\mathbf{A}_1}$ is a trapdoor for $\Lambda_q^{\perp}((\mathcal{H}(\mathcal{G}(\mathsf{id})) - \mathcal{H}(\mathcal{G}(\mathsf{id}^*)))\mathbf{A}_1)$. $\mathcal{C}_1$ can respond a long-term private key query for any $\mathsf{id} \neq \mathsf{id}^*$ by running $\mathsf{SampleRight}(\mathbf{A}_{\mathsf{id}}, (\mathcal{H}(\mathcal{G}(\mathsf{id})) - \mathcal{H}(\mathcal{G}(\mathsf{id}^*)))\mathbf{A}_1, \mathbf{R}_1^*, \mathbf{R}_{\mathbf{A}_1}, \mathbf{v}, s)$ to generate a short vector $\mathbf{e}_{\mathsf{id}} \in \mathbb{Z}^{3m}$. The parameter $s = \widetilde{\mathcal{O}}(m)$ is sufficiently large, and according to Lemma 7, $\mathbf{e}_{\mathsf{id}}$ is statistically close to that in Game 1. The remainders are unchanged and identical to those in Game 1. $\mathbf{A}$ and $\mathbf{A}_1$ are statistically close to those in Game 1, the advantage of $\mathcal{A}_1$ in Game 2 is at most negligibly different from that in Game 1.

Game 3: We redesign the partial challenge ciphertexts $\mathbf{c}_1^*$ and $\mathbf{c}_0'^*$, and the remainders (including $\mathbf{c}_{2,\theta}^*$) are unchanged and identical to those in Game 2. $\mathcal{C}_1$ first samples $c_0^* \xleftarrow{\$} \mathbb{Z}_q$ and $\mathbf{c}_1^* \xleftarrow{\$} \mathbb{Z}_q^{3m}$, then set $\mathbf{c}_0'^* = \mathbf{U}^{\mathrm{T}}\mathbf{s}_0' + \mathbf{e}_0' + \mathsf{bin}(c_0^*)\lfloor \frac{q}{2} \rfloor \bmod q \in \mathbb{Z}_q^k$. Because c_0^* and $\mathbf{c}_1^*$ are always random, the advantage of $\mathcal{A}_1$ in returning a correct c_0^* is zero, and the same advantage zero for the message m is returned correctly.

A reduction from the LWE problem will be given to show that Game 2 and Game 3 are computationally indistinguishable for a ppt adversary.

A reduction from LWE: Assume that there is a ppt $\mathcal{A}_1$ distinguishing Games 2 and 3 with non-negligible advantage, then we use $\mathcal{A}_1$ to design an algorithm $\mathcal{B}_1$ to solve the LWE problem defined in Definition 4.

Given an LWE instance, a fresh pair $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for $i = 1, 2, \cdots, m+1$, from a sampling oracle, which is truly random $\mathcal{R}_{\$}$ or noisy pseudo-random $\mathcal{R}_{\mathbf{s}_0}$ for a secret vector $\mathbf{s}_0 \in \mathbb{Z}_q^n$. The target of $\mathcal{B}_1$ is to distinguish between the two oracles by utilizing $\mathcal{A}_1$ as follows:

Instance: $\mathcal{B}_1$ receives an LWE instance (i.e., $(\mathbf{a}_i, b_i), i = 1, 2, \cdots, m+1$).

Setup: $\mathcal{B}_1$ does as follows:

1. Assemble $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ from m of the given samples, define $\mathbf{A} = [\mathbf{a}_1|\cdots|\mathbf{a}_m]$.
2. Assemble $\mathbf{v} \in \mathbb{Z}_q^n$ from the unused samples, that is, define $\mathbf{v} = \mathbf{a}_{m+1}$.
3. Run $\mathsf{TrapGen}(q,n,m)$ to generate $\mathbf{A}_1 \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{R}_{\mathbf{A}_1}$, $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{R}_{\mathbf{B}}$, sample $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$, $\mathbf{B}_0, \mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
4. Design $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$, as in Game 2 by using id^* and R_1^*.
5. Sample a collision-resistance hash function $\mathcal{G} : \{0,1\}^* \to \mathbb{Z}_q^n$, and an FRD function $\mathcal{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$.
6. Let $\mathsf{pp} = (\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{B}_0, \mathbf{v}, \mathbf{B}_1, \mathbf{U}, \mathcal{G}, \mathcal{H})$, and send pp to $\mathcal{A}_1$.

Queries: $\mathcal{B}_1$ answers a long-term private key query for $\mathsf{id} \neq \mathsf{id}^*$ as in Game 2. As $\mathcal{B}_1$ knows the master secret key $\mathbf{R}_{\mathbf{B}}$, it can answer a time update key query for any t (a list of shorter matrices $(\mathbf{E}_\theta)_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$) as in the real game.

Challenge: Once receive m^*, $\mathcal{B}_1$ computes a challenge for id^* and t^* as follows:

1. Assemble $\mathbf{e}_1^*$ from m of the samples, define $\mathbf{e}_1^* = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$, and $v^* = b_{m+1}$.

2. Blind m^* by defining $c_0^* = v^* + \mathsf{m}^* \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$, and $\mathbf{c}_1^* = \begin{bmatrix} \mathbf{e}_1^* \\ \mathbf{R}_1^{*\mathrm{T}} \mathbf{e}_1^* \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
3. For $\theta \in \mathsf{path}(\mathsf{id}^*)$, define $\mathbf{B}_{\mathsf{id}_\theta^*, \mathsf{t}^*} = [\mathbf{B} | \mathbf{B}_0 + \mathcal{H}(\mathcal{G}(\mathsf{t}^* || \theta)) \mathbf{B}_1] \in \mathbb{Z}_q^{n \times 2m}$.
4. Sample $\mathbf{s}_0' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_0' \xleftarrow{\$} \chi^k$, $\mathbf{e}_1' \xleftarrow{\$} \chi^m$, and $\mathbf{R}_2 \xleftarrow{\$} \{1, -1\}^{m \times m}$.
5. Let $\mathbf{c}_0'^* = \mathbf{U}^{\mathrm{T}} \mathbf{s}_0' + \mathbf{e}_0' + \mathsf{bin}(c_0^*) \lfloor \frac{q}{2} \rfloor$, and $\mathbf{c}_{2,\theta}^* = \mathbf{B}_{\mathsf{id}_\theta^*, \mathsf{t}}^{\mathrm{T}} \mathbf{s}_0' + \begin{bmatrix} \mathbf{e}_1' \\ \mathbf{R}_2^{\mathrm{T}} \mathbf{e}_1' \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
6. Send $\mathsf{ct}_{\mathsf{id}^*, \mathsf{t}^*} = (\mathsf{id}^*, \mathsf{t}^*, \mathbf{c}_0'^*, \mathbf{c}_1^*, (\mathbf{c}_{2,\theta}^*)_{\theta \in \mathsf{path}(\mathsf{id}^*)})$ to $\mathcal{A}_1$.

Obviously, c_0^* can be derived from $(\mathbf{c}_0'^*, (\mathbf{c}_{2,\theta}^*)_{\theta \in \mathsf{path}(\mathsf{id}^*)})$ by using a time update key $(\mathbf{E}_\theta)_{\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}^*)}$. We first show that if the LWE instance is from a noisy pseudo-random $\mathcal{R}_{\mathbf{s}_0}$, so $(c_0^*, \mathbf{c}_1^*)$ enjoys a distribution exactly as in Game 2. First, it can be checked that $\mathbf{A}_{\mathsf{id}^*} = [\mathbf{A} | \mathbf{A}\mathbf{R}_1^*] \in \mathbb{Z}_q^{n \times 2m}$. Second, it can be checked that $\mathbf{e}_1^* = \mathbf{A}^{\mathrm{T}} \mathbf{s}_0 + \mathbf{e}_1 \bmod q$, where $\mathbf{e}_1 \xleftarrow{\$} \chi^m$. Thus, $\mathbf{c}_1^*$ enjoys the following structure:

$$\mathbf{c}_1^* = \begin{bmatrix} c\mathbf{e}_1^* \\ \mathbf{R}_1^{*\mathrm{T}} \mathbf{e}_1^* \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{\mathrm{T}} \mathbf{s}_0 + \mathbf{e}_1 \\ (\mathbf{A}\mathbf{R}_1^*)^{\mathrm{T}} \mathbf{s}_0 + \mathbf{R}_1^{*\mathrm{T}} \mathbf{e}_1 \end{bmatrix} = \mathbf{A}_{\mathsf{id}^*}^{\mathrm{T}} \mathbf{s}_0 + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{R}_1^{*\mathrm{T}} \mathbf{e}_1 \end{bmatrix} \bmod q,$$

which implies that $\mathbf{c}_1^*$ is exactly the $\mathbf{c}_1$ part of a valid challenge in Game 2.

We then show that if the LWE instance is from a truly random $\mathcal{R}_\$$, so $(c_0^*, \mathbf{c}_1^*)$ enjoys a distribution exactly as in Game 3. It can be checked that v^* is unform over $\mathbb{Z}_q$, and $\mathbf{e}_1^*$ is unform over $\mathbb{Z}_q^m$. Thus, $\mathbf{c}_1^*$ is unform and independent over $\mathbb{Z}_q^{2m}$, which implies that $\mathbf{c}_1^*$ is exactly the $\mathbf{c}_1$ part of a valid challenge ciphertext in Game 3.

Guess: After making some additional queries, $\mathcal{A}_1$ returns a guess for which challenger, Game 2 or Game 3, it is interacting with. Then, $\mathcal{B}_1$ returns the guess of $\mathcal{A}_1$ as an answer to the given LWE instance.

According to the above analysis, if the LWE instance is from $\mathcal{R}_{\mathbf{s}_0}$, $\mathcal{A}_1$'s view is as in Game 2, and if the LWE instance is from $\mathcal{R}_\$$, $\mathcal{A}_1$'s view is as in Game 3, and thus, the advantage of $\mathcal{B}_1$ in solving the LWE problem is the same as that of $\mathcal{A}_1$ in distinguishing Game 2 and Game 3. This completes the proof.

4 Our Lattice-Based RIBE-CE Scheme with DKER

Our RIBE-CE scheme with DKER in the standard model is a combination of a two-level lattice-based HIBE scheme and an IBE scheme [1] from which the PKG still issues a long-term private key to each system user id, yet this private key is a trapdoor matrix, not a shorter vector as in our first scheme. This trapdoor matrix ensures each user id computing a short-term decryption key (a shorter vector) for any time period on their own. Similarly, the BT revocation mechanism is adopted to alleviate the workload of PKG.

4.1 Description of the Scheme

As in our first scheme, our lattice-based RIBE-CE with DKER in the standard model consists of eight pt algorithms: Setup, PriKeyGen, KeyUpd, DecKeyGen, Encrypt, Evolve, Decrypt and Revoke. The algorithms are described as follows:

– $\mathsf{Setup}(1^n, N)$: On input a security parameter n and the maximal number of users $N = 2^n$, parameters q, m, k, s and $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$ are the same as in our first scheme. PKG specifies the same steps as in our first scheme except it additionally samples $\mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and thus:
  1. Set $\mathsf{pp} = (\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \mathbf{v}, \mathbf{U}, \mathcal{G}, \mathcal{H})$, and $\mathsf{msk} = (\mathbf{R}_\mathbf{A}, \mathbf{R}_\mathbf{B})$.
  2. Output $(\mathsf{pp}, \mathsf{msk}, \mathsf{RL}, \mathsf{st})$ where msk is kept in *secret* by PKG, and pp is made *public* and as an implicit input of all other algorithms.
– $\mathsf{PriKeyGen}(\mathsf{msk}, \mathsf{id}, \mathsf{st})$: On input an identity $\mathsf{id} \in \mathcal{I}$, the master secret key msk and the state st. PKG specifies the following steps:
  1. View id as an unassigned leaf node of BT, thus, $\mathsf{id} \in 0||\{0,1\}^n$.
  2. Define $\mathbf{A}_\mathsf{id} = [\mathbf{A}|\mathbf{A}_0 + \mathcal{H}(\mathcal{G}(\mathsf{id}))\mathbf{A}_2] \in \mathbb{Z}_q^{n\times 2m}$.
  3. Run $\mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{R}_\mathbf{A}, \mathbf{A}_\mathsf{id}), s)$ to generate a trapdoor $\mathbf{R}_{\mathbf{A}_\mathsf{id}}$ for $\Lambda_q^\perp(\mathbf{A}_\mathsf{id})$.
  4. Output an updated state st, and $\mathsf{sk}_\mathsf{id} = \mathbf{R}_{\mathbf{A}_\mathsf{id}}$.
– $\mathsf{KeyUpd}(\mathsf{RL}, \mathsf{t}, \mathsf{msk}, \mathsf{st})$: The same as in our first scheme.
– $\mathsf{DecKeyGen}(\mathsf{sk}_\mathsf{id}, \mathsf{uk}_\mathsf{t}, \mathsf{t})$: On input a long-term private key $\mathsf{sk}_\mathsf{id} = \mathbf{R}_{\mathbf{A}_\mathsf{id}}$, a time t and current time update key $\mathsf{uk}_\mathsf{t} = (\theta, \mathbf{E}_\theta)_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$. The recipient id specifies the following steps:
  1. If $\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}) = \emptyset$, return $\perp$ and abort.
  2. Otherwise, define $\mathbf{A}_{\mathsf{id},\mathsf{t}} = [\mathbf{A}_\mathsf{id}|\mathbf{A}_1 + \mathcal{H}(\mathcal{G}(\mathsf{t}))\mathbf{A}_2] \in \mathbb{Z}_q^{n\times 3m}$.
  3. Run $\mathsf{SampleLeft}(\mathbf{A}_{\mathsf{id},\mathsf{t}}, \mathbf{R}_{\mathbf{A}_\mathsf{id}}, \mathbf{v}, s)$ to generate $\mathbf{e}_{\mathsf{id},\mathsf{t}} \in \mathbb{Z}^{3m}$ satisfying $\mathbf{A}_{\mathsf{id},\mathsf{t}} \cdot \mathbf{e}_{\mathsf{id},\mathsf{t}} = \mathbf{v} \bmod q$.
  4. Select $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$, and return $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = (\mathbf{E}_\theta, \mathbf{e}_{\mathsf{id},\mathsf{t}})$.
– $\mathsf{Encrypt}(\mathsf{id}, \mathsf{t}, \mathsf{m})$: On input an identity $\mathsf{id} \in \mathcal{I}$, a time $\mathsf{t} \in \mathcal{T}$, and a message $\mathsf{m} \in \{0,1\}$. The sender will specify the following steps:
  1. Let $\mathbf{A}_{\mathsf{id},\mathsf{t}} = [\mathbf{A}|\mathbf{A}_0 + \mathcal{H}(\mathcal{G}(\mathsf{id}))\mathbf{A}_2|\mathbf{A}_1 + \mathcal{H}(\mathcal{G}(\mathsf{t}))\mathbf{A}_2] \in \mathbb{Z}_q^{n\times 3m}$.
  2. For $\theta \in \mathsf{path}(\mathsf{id})$, define $\mathbf{B}_{\mathsf{id}_\theta,\mathsf{t}} = [\mathbf{B}|\mathbf{B}_0 + \mathcal{H}(\mathcal{G}(\mathsf{t}||\theta))\mathbf{B}_1] \in \mathbb{Z}_q^{n\times 2m}$.
  3. Sample $\mathbf{s}_0, \mathbf{s}_0' \xleftarrow{\$} \mathbb{Z}_q^n$, $e_0 \xleftarrow{\$} \chi$, $\mathbf{e}_0' \xleftarrow{\$} \chi^k$, $\mathbf{e}_1, \mathbf{e}_1' \xleftarrow{\$} \chi^m$, and $\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2 \xleftarrow{\$} \{1,-1\}^{m\times m}$.
  4. Let $c_0 = \mathbf{v}^\mathrm{T}\mathbf{s}_0 + e_0 + \mathsf{m}\lfloor \frac{q}{2} \rfloor \bmod q \in \mathbb{Z}_q$, $\mathbf{c}_1 = \mathbf{A}_{\mathsf{id},\mathsf{t}}^\mathrm{T}\mathbf{s}_0 + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{R}_0^\mathrm{T}\mathbf{e}_1 \\ \mathbf{R}_1^\mathrm{T}\mathbf{e}_1 \end{bmatrix} \in \mathbb{Z}_q^{3m}$.
  5. Let $\mathbf{c}_0' = \mathbf{U}^\mathrm{T}\mathbf{s}_0' + \mathbf{e}_0' + \mathsf{bin}(c_0)\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^k$, and $\mathbf{c}_{2,\theta} = \mathbf{B}_{\mathsf{id}_\theta,\mathsf{t}}^\mathrm{T}\mathbf{s}_0' + \begin{bmatrix} \mathbf{e}_1' \\ \mathbf{R}_2^\mathrm{T}\mathbf{e}_1' \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
  6. Output $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})}) \in (0||\{0,1\}^n)^2 \times \mathbb{Z}_q^k \times \mathbb{Z}_q^{3m} \times (\mathbb{Z}_q^{2m})^{n+1}$.
– $\mathsf{Evolve}(\mathsf{ct}_{\mathsf{id},\mathsf{t}}, \mathsf{t}', \mathsf{uk}_\mathsf{t})$: On input an original ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})})$ or an evolutive ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}'', \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})})$, a new time $\mathsf{t}' > \mathsf{t} > \mathsf{t}''$, and the current time update key $\mathsf{uk}_\mathsf{t} = (\theta, \mathbf{E}_\theta)_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{RL},\mathsf{t})}$. The Cloud specifies the following steps:
  1. If the recipient has been revoked before (or at) time t, set $\mathsf{ct}_{\mathsf{id},\mathsf{t}'} = \mathsf{ct}_{\mathsf{id},\mathsf{t}}$.
  2. Otherwise, compute $\mathbf{w}_0 = \mathbf{c}_0' - \mathbf{E}_\theta^\mathrm{T} \cdot \mathbf{c}_{2,\theta} \bmod q \in \mathbb{Z}_q^k$, here, $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
  3. Compute $c_0 = (1, 2, \cdots, 2^{k-1}) \cdot \lfloor \frac{2}{q}\mathbf{w}_0 \rfloor \in \mathbb{Z}_q$.
  4. Sample $\mathbf{s}_0'' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_0'' \xleftarrow{\$} \chi^k$, $\mathbf{e}_1'' \xleftarrow{\$} \chi^m$, and $\mathbf{R}_2' \xleftarrow{\$} \{1,-1\}^{m\times m}$.

5. For $\theta \in \mathsf{path}(\mathsf{id})$, define $\mathbf{B}_{\mathsf{id}_\theta,\mathsf{t}'} = [\mathbf{B}|\mathbf{B}_0 + \mathcal{H}(\mathcal{G}(\mathsf{t}'||\theta))\mathbf{B}_1] \in \mathbb{Z}_q^{n\times 2m}$.
6. Let $\mathbf{c}_0'' = \mathbf{U}^{\mathrm{T}}\mathbf{s}_0'' + \mathbf{e}_0'' + \mathsf{bin}(c_0)\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^k$, $\mathbf{c}_{2,\theta}' = \mathbf{B}_{\mathsf{id}_\theta,\mathsf{t}'}^{\mathrm{T}}\mathbf{s}_0'' + \begin{bmatrix} \mathbf{e}_1'' \\ \mathbf{R}_2'^{\mathrm{T}}\mathbf{e}_1'' \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
7. Output $\mathsf{ct}_{\mathsf{id},\mathsf{t}'} = (\mathsf{id}, \mathsf{t}(\text{or } \mathsf{t}''), \mathsf{t}', \mathbf{c}_0'', \mathbf{c}_1, (\mathbf{c}_{2,\theta}')_{\theta\in\mathsf{path}(\mathsf{id})}) \in (0||\{0,1\}^n)^3 \times \mathbb{Z}_q^k \times \mathbb{Z}_q^{3m} \times (\mathbb{Z}_q^{2m})^{n+1}$.

– $\mathsf{Decrypt}(\mathsf{dk}_{\mathsf{id}',\mathsf{t}'}, \mathsf{ct}_{\mathsf{id},\mathsf{t}})$: On input an original ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})})$ or an evolutive ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}'', \mathsf{t}, \mathbf{c}_0'', \mathbf{c}_1, (\mathbf{c}_{2,\theta}')_{\theta\in\mathsf{path}(\mathsf{id})})$, and a decryption key $\mathsf{dk}_{\mathsf{id}',\mathsf{t}'}$. The recipient id' needs to specify the following steps:

1. For the original ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})})$:
   1.1. If $(\mathsf{id} \neq \mathsf{id}') \vee (\mathsf{t} \neq \mathsf{t}')$, return $\perp$ and abort.
   1.2. Otherwise, compute $\mathbf{w}_0 = \mathbf{c}_0' - \mathbf{E}_\theta^{\mathrm{T}} \cdot \mathbf{c}_{2,\theta} \bmod q \in \mathbb{Z}_q^k$, here, $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
   1.3. Compute $c_0 = (1, 2, \cdots, 2^{k-1}) \cdot \lfloor \frac{2}{q}\mathbf{w}_0 \rceil \in \mathbb{Z}_q$, $w = c_0 - \mathbf{e}_{\mathsf{id},\mathsf{t}}^{\mathrm{T}}\mathbf{c}_1 \in \mathbb{Z}_q$.
   1.4. Output $\lfloor \frac{2}{q}w \rceil \in \{0,1\}$.
2. For the evolutive ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}'', \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})})$:
   2.1. If $(\mathsf{id} \neq \mathsf{id}') \vee (\mathsf{t} \neq \mathsf{t}')$, return $\perp$ and abort.
   2.2. Otherwise, compute $\mathbf{w}_0 = \mathbf{c}_0'' - \mathbf{E}_\theta^{\mathrm{T}} \cdot \mathbf{c}_{2,\theta}' \bmod q \in \mathbb{Z}_q^k$, here, $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
   2.3. Compute $c_0 = (1, 2, \cdots, 2^{k-1}) \cdot \lfloor \frac{2}{q}\mathbf{w}_0 \rceil \in \mathbb{Z}_q$.
   2.4. Define $\mathbf{A}_{\mathsf{id},\mathsf{t}''} = [\mathbf{A}_{\mathsf{id}}|\mathbf{A}_1 + \mathcal{H}(\mathcal{G}(\mathsf{t}''))\mathbf{A}_2] \in \mathbb{Z}_q^{n\times 3m}$.
   2.5. Run $\mathsf{SampleLeft}(\mathbf{A}_{\mathsf{id},\mathsf{t}''}, \mathbf{R}_{\mathbf{A}_{\mathsf{id}}}, \mathbf{v}, s)$ to generate $\mathbf{e}_{\mathsf{id},\mathsf{t}''} \in \mathbb{Z}^{3m}$ satisfying $\mathbf{A}_{\mathsf{id},\mathsf{t}''} \cdot \mathbf{e}_{\mathsf{id},\mathsf{t}''} = \mathbf{v} \bmod q$.
   2.6. Define $w = c_0 - \mathbf{e}_{\mathsf{id},\mathsf{t}''}^{\mathrm{T}}\mathbf{c}_1 \bmod q \in \mathbb{Z}_q$.
   2.7. Output $\lfloor \frac{2}{q}w \rceil \in \{0,1\}$.

– $\mathsf{Revoke}(\mathsf{id}, \mathsf{t}, \mathsf{RL}, \mathsf{st})$: The same as in our first scheme.

4.2 Analysis

We analysis the efficiency, correctness and security of our lattice-based RIBE-CE scheme with DKER in the standard model.

Efficiency: The efficiency aspect of our lattice-based RIBE-CE scheme with DKER in the standard model and $N = 2^n$ is as follows:

– The bit-size of public parameters pp is $(7nm + n + nk + 2n)\log q = \widetilde{\mathcal{O}}(n^2)$.
– The long-term private key $\mathsf{sk}_{\mathsf{id}}$ has a trapdoor matrix of bit-size $\widetilde{\mathcal{O}}(n^2)$.
– The time update key uk_{t} has bit-size $\mathcal{O}(r \log \frac{N}{r}) \cdot \widetilde{\mathcal{O}}(n)$ where r is the number of revoked users.
– The ciphertext $\mathsf{ct}_{\mathsf{id},\mathsf{t}}$ has bit-size $2(n+1) + (k + 3m + 2m(n+1))\log q = \widetilde{\mathcal{O}}(n^2)$.
– The short-term decryption key $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$ has bit-size $\widetilde{\mathcal{O}}(n)$.

Correctness: If the first lattice-based RIBE-CE with DKER in the standard model is operated correctly as specified, and a recipient id is not revoked at time t, then $\mathsf{dk}_{\mathsf{id},\mathsf{t}} = (\mathbf{E}_\theta, \mathbf{e}_{\mathsf{id},\mathsf{t}})$ satisfies $\mathbf{B}_{\mathsf{t}_\theta} \cdot \mathbf{E}_\theta = \mathbf{U} \bmod q$ and $\mathbf{A}_{\mathsf{id},\mathsf{t}} \cdot \mathbf{e}_{\mathsf{id},\mathsf{t}} = \mathbf{v} \bmod q$. In the decryption algorithm, the non-revoked id tries to derive m by using $\mathsf{dk}_{\mathsf{id},\mathsf{t}}$ (sometimes, id also needs to use the long-term private key to derive a new shorter vector according to a new time):

– If the given ciphertext is an original ciphertext, $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}, \mathbf{c}_0', \mathbf{c}_1, (\mathbf{c}_{2,\theta})_{\theta\in\mathsf{path}(\mathsf{id})})$.

1. Parse $\mathbf{c}_{2,\theta} = \begin{bmatrix}\mathbf{c}_{2,0}\\ \mathbf{c}_{2,1}\end{bmatrix}$ where $\mathbf{c}_{2,i\in\{0,1\}} \in \mathbb{Z}_q^m$ and $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
2. Compute
$\mathbf{w}_0 = \mathbf{c}_0' - \mathbf{E}_\theta^{\mathrm{T}}\mathbf{c}_{2,\theta} = \mathbf{U}^{\mathrm{T}}\mathbf{s}_0' + \mathbf{e}_0' + \mathsf{bin}(c_0)\lfloor\frac{q}{2}\rfloor - \mathbf{E}_\theta^{\mathrm{T}}\begin{bmatrix}\mathbf{c}_{2,0}\\ \mathbf{c}_{2,1}\end{bmatrix} = \mathsf{bin}(c_0)\lfloor\frac{q}{2}\rfloor + \underbrace{\mathbf{e}_0' - \mathbf{E}_\theta^{\mathrm{T}}\begin{bmatrix}\mathbf{e}_1'\\ \mathbf{R}_2^{\mathrm{T}}\mathbf{e}_1'\end{bmatrix}}_{\mathsf{error}'}$

   According to our parameters settings, it can be checked that the error term error' is bounded by $q/5$ (i.e., $\|\mathsf{error}'\|_\infty < q/5$), thus, we have the conclusion $(1, 2, \cdots, 2^{k-1}) \cdot \lfloor\frac{2}{q}\mathbf{w}_0\rceil = c_0$ with overwhelming probability.
3. Parse $\mathbf{c}_1 = \begin{bmatrix}\mathbf{c}_{1,0}\\ \mathbf{c}_{1,1}\\ \mathbf{c}_{1,2}\end{bmatrix}$ where $\mathbf{c}_{1,i\in\{0,1,2\}} \in \mathbb{Z}_q^m$, and compute
$w = c_0 - \mathbf{e}_{\mathsf{id},\mathsf{t}}^{\mathrm{T}}\mathbf{c}_1 = \mathbf{v}^{\mathrm{T}}\mathbf{s}_0 + e_0 + \mathsf{m}\lfloor\frac{q}{2}\rfloor - \mathbf{e}_{\mathsf{id},\mathsf{t}}^{\mathrm{T}}\begin{bmatrix}\mathbf{c}_{1,0}\\ \mathbf{c}_{1,1}\\ \mathbf{c}_{1,2}\end{bmatrix} = \mathsf{m}\lfloor\frac{q}{2}\rfloor + \underbrace{e_0 - \mathbf{e}_{\mathsf{id},\mathsf{t}}^{\mathrm{T}}\begin{bmatrix}\mathbf{e}_1\\ \mathbf{R}_0^{\mathrm{T}}\mathbf{e}_1\\ \mathbf{R}_1^{\mathrm{T}}\mathbf{e}_1\end{bmatrix}}_{\mathsf{error}}$

   According to our parameters settings, it can be checked that the error term error is bounded by $q/5$ (i.e., $\|\mathsf{error}\|_\infty < q/5$), thus, we have the conclusion $\lfloor\frac{2}{q}w\rceil = \mathsf{m}$ with overwhelming probability.

– If the given ciphertext is an evolutive ciphertext, $\mathsf{ct}_{\mathsf{id},\mathsf{t}} = (\mathsf{id}, \mathsf{t}'', \mathsf{t}, \mathbf{c}_0'', \mathbf{c}_1, (\mathbf{c}_{2,\theta}')_{\theta\in\mathsf{path}(\mathsf{id})})$,

1. Parse $\mathbf{c}_{2,\theta}' = \begin{bmatrix}\mathbf{c}_{2,0}'\\ \mathbf{c}_{2,1}'\end{bmatrix}$ where $\mathbf{c}_{2,i\in\{0,1\}}' \in \mathbb{Z}_q^m$ and $\theta \in (\mathsf{path}(\mathsf{id}) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{RL}, \mathsf{t}))$.
2. Compute
$\mathbf{w}_0 = \mathbf{c}_0'' - \mathbf{E}_\theta^{\mathrm{T}}\boldsymbol{c}_{2,\theta}' = \mathbf{U}^{\mathrm{T}}\mathbf{s}_0'' + \mathbf{e}_0'' + \mathsf{bin}(c_0)\lfloor\frac{q}{2}\rfloor - \mathbf{E}_\theta^{\mathrm{T}}\begin{bmatrix}\mathbf{c}_{2,0}'\\ \mathbf{c}_{2,1}'\end{bmatrix} = \mathsf{bin}(c_0)\lfloor\frac{q}{2}\rfloor + \underbrace{\mathbf{e}_0'' - \mathbf{E}_\theta^{\mathrm{T}}\begin{bmatrix}\mathbf{e}_1''\\ \mathbf{R}_2'^{\mathrm{T}}\mathbf{e}_1''\end{bmatrix}}_{\mathsf{error}''}$

   According to our parameters settings, it can be checked that error'' is bounded by $q/5$ (i.e., $\|\mathsf{error}''\|_\infty < q/5$), thus, we have the conclusion $(1, 2, \cdots, 2^{k-1}) \cdot \lfloor\frac{2}{q}\mathbf{w}_0\rceil = c_0$ with overwhelming probability.
3. Let $\mathbf{A}_{\mathsf{id},\mathsf{t}''} = [\mathbf{A}_{\mathsf{id}} | \mathbf{A}_1 + \mathcal{H}(\mathcal{G}_1(\mathsf{t}''))\mathbf{A}_2] \in \mathbb{Z}_q^{n\times 3m}$.
4. Run $\mathsf{SampleLeft}(\mathbf{A}_{\mathsf{id},\mathsf{t}''}, \mathbf{R}_{\mathbf{A}_{\mathsf{id}}}, \mathbf{v}, s)$ to generate $\mathbf{e}_{\mathsf{id},\mathsf{t}''} \in \mathbb{Z}^{3m}$ satisfying $\mathbf{A}_{\mathsf{id},\mathsf{t}''} \cdot \mathbf{e}_{\mathsf{id},\mathsf{t}''} = \mathbf{v} \bmod q$.
5. Parse $\mathbf{c}_1 = \begin{bmatrix}\mathbf{c}_{1,0}\\ \mathbf{c}_{1,1}\\ \mathbf{c}_{1,2}\end{bmatrix}$ where $\mathbf{c}_{1,i\in\{0,1,2\}} \in \mathbb{Z}_q^m$, and compute
$w = c_0 - \mathbf{e}_{\mathsf{id},\mathsf{t}''}^{\mathrm{T}}\mathbf{c}_1 = \mathbf{v}^{\mathrm{T}}\mathbf{s}_0 + e_0 + \mathsf{m}\lfloor\frac{q}{2}\rfloor - \mathbf{e}_{\mathsf{id},\mathsf{t}''}^{\mathrm{T}}\begin{bmatrix}\mathbf{c}_{1,0}\\ \mathbf{c}_{1,1}\\ \mathbf{c}_{1,2}\end{bmatrix} = \mathsf{m}\lfloor\frac{q}{2}\rfloor + \underbrace{e_0 - \mathbf{e}_{\mathsf{id},\mathsf{t}''}^{\mathrm{T}}\begin{bmatrix}\mathbf{e}_1\\ \mathbf{R}_0^{\mathrm{T}}\mathbf{e}_1\\ \mathbf{R}_1^{\mathrm{T}}\mathbf{e}_1\end{bmatrix}}_{\mathsf{error}}$

   According to our parameters settings, it can be checked that the error term error is bounded by $q/5$ (i.e., $\|\mathsf{error}\|_\infty < q/5$), thus, we have the conclusion $\lfloor\frac{2}{q}w\rceil = \mathsf{m}$ with overwhelming probability.

Theorem 2. *Our* RIBE-CE *scheme with* DKER *in the standard model is* ind-cpa *secure if the* LWE *assumption holds.*

Proof. The proof is similar to that in Theorem 1, and due to the limited space, the details are presented in the full paper.

5 Conclusion

In this paper, we propose two (and the first) lattice-based RIBE schemes with server-aided ciphertext evolution. In comparison with previous lattice-based constructions of RIBE, our two schemes enjoy a significant advantage in terms of ciphertext security when considering the scenario of a secure data (i.e., ciphertext) sharing in the cloud setting and the revoked users cannot access to both the previously and subsequently shared data. The BT revocation mechanism is adopted for time key update, thus our three schemes only obtain a logarithmic complexity workload of the PKG, which serves as one solution to the challenge posed by Sun et al.. In particular, we remedy the security model and introduce DKER property into RIBE-CE, a default security requirement for RIBE, which has not been considered by Sun et al.. Our first and second lattice-based RIBE-CE schemes are without DKER in the standard model and with DKER in the standard model, respectively.

Acknowledgments. The authors would like to thank the anonymous reviewers of Inscrypt 2021 for their helpful comments and this research was supported by National Natural Science Foundation of China (Grant No. 61802075), Guangxi key Laboratory of Cryptography and Information Security (Grant No. GCIS201907) and Natural Science Foundation of Henan Province (Grant No. 202300410508.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010) https://doi.org/10.1007/978-3-642-13190-5_28
2. Ajtai, M.: Generating Hard Instances of Lattice Problems (Extended Abstract). In: STOC, pp. 99–108. ACM (1996) https://doi.org/10.1145/237814.237838
3. Alwen, J., Peikert, C.: Generating Shorter Bases for Hard Random Lattices. Theor. Comput. Sys. **48**(3), 535–553 (2011) https://doi.org/10.1007/s00224-010-9278-3
4. Boldyreva, A., Goyal, V., Kumar, V.: Identity-Based Encryption with Efficient Revocation. In: CCS, pp. 417–426. ACM (2008) https://doi.org/10.1145/1455770.1455823
5. Boneh, D., Farnklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001) https://doi.org/10.1007/3-540-44647-8_13
6. Cash, D., Hofheinzy, D., Kiltz, E., et al.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010) https://doi.org/10.1007/978-3-642-13190-5_27
7. Chen, J., Lim, H., Ling, S., et al.: Revocable Identity-Based Encryption from Lattices. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 390–403. Springer, Heidelberg (2012) https://doi.org/10.1007/978-3-642-31448-3_29
8. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001) https://doi.org/10.1007/3-540-45325-3_32

9. Dong, C., Yang, K., Qiu, J., et al.: Outsouraced Revocable Identity-Based Encryption from Lattices. Trans. Emerging Tel. Tech. e3529 (2018) https://doi.org/10.1002/ett.3529
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoor for Hard Lattices and New Cryptographic Constructions. In: STOC, pp. 197–206. ACM (2008) https://doi.org/10.1145/1374376.1374407
11. Katsumata, S., Matsuda, T., Takayasu, A.: Lattice-Based Revocable (Hierarchical) IBE with Decryption Key Exposure Resistance. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11443, pp. 441–471. Springer, Cham (2019) https://doi.org/10.1007/978-3-030-17259-6_15
12. Lee, K.: A Generic Construction for Revocable Identity-Based Encryption with Subset Difference Methods. PLOS ONE **15**(9), e0239053 (2020) https://doi.org/10.1371/journal.pone.o239053
13. Li, J., Li, J., Chen, X., et al.: Identity-Based Encryption with Outsourced Revocation in Cloud Computing. IEEE Trans. Comput. **64**(2), 426–437 (2015) https://doi.org/10.1109/TC.2013.208
14. Liang, K., Liu, J., Wong, D., et al.: An Efficient Cloud-Based Revocable Identity-Based Procy Re-Rncryption Scheme for Public Clouds Data Sharing. In: Kutyowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 257–272. Springer, Cham (2014) https://doi.org/10.1007/978-3-319-11203-9_15
15. Libert B, Vergnaud D. Adaptive-ID Secure Revocable Identity-Based Encryption. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 1–15. Springer, Heidelberg (2009) https://doi.org/10.1007/978-3-642-00862-7_1
16. Ma, X., Lin, D.: Generic Constructions of Revocable Identity-Based Encryption. In: Liu, Z., Yung, M. (eds.) INSCRYPT 2019. LNCS, vol. 12020, pp. 381–396. Springer, Cham (2019) https://doi.org/10.1007/978-3-030-42921-8_22
17. Micciancio, D,, Peiker,t C.: Hardness of SIS and LWE with Small Parameters. In: Canetti, R., Garay, J.A.. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 21–39. Springer, Heidelberg (2013) https://doi.org/10.1007/978-3-642-40041-4_2
18. Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012) https://doi.org/10.1007/978-3-642-29011-4_41
19. Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2010. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001) https://doi.org/10.1007/3-540-44647-8_3
20. Nguyen, K., Wang, H., Zhang, J.: Server-Aided Revocable Identity-Based Encryption from Lattices. In: Foresti, S., Persiano, G. (eds.) CANS 2010. LNCS, vol. 10052, pp. 107–123. Springer, Cham (2016) https://doi.org/10.1007/978-3-319-48965-0_7
21. Qin, B., Deng, R., Li, Y., et al.: Server-Aided Revocable Identity-Based Encryption. In: Pernul, G., Ryan, P., Weippl, E. (eds.) ESORICS 2015. LNCS, vol. 9326, pp. 286–304. Springer, Cham (2015) https://doi.org/10.1007/978-3-319-24174-6_15
22. Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In: STOC, pp. 84–93. ACM (2005) https://doi.org/10.1145/1060590.1060603
23. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 98–115. Springer, Heidelberg (2005) https://doi.org/10.1007/11426639_27

24. Seo, J., Emura, K.: Revocable Identity-Based Encryption Revisited: Security and Construction. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 216–234. Springer, Heidelberg (2013)
https://doi.org/10.1007/978-3-642-36362-7_14
25. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
https://doi.org/10.1007/978-3-642-14623-7_6
26. Shor, P.: Polynomial-Time Algorithms for Prime Factorization and Dislogarithms on A Quantum Computer. SIAN Journal on Computing **26**(5), 1485–1509 (1997)
https://doi.org/10.1016/j.tcs.2020.02.03
27. Sun, Y., Mu, Y., Susilo, W., et al.: Revocable Identity-Based Encryption with Server-Aided Ciphertext Evolution. Theor. Comput. Sci. **2020**(815), 11–24 (2020)
https://doi.org/10.1016/j.tcs.2020.02.03
28. Takayasu, A.: Adaptively Secure Lattice-Based Revocable IBE in the QROM: Compact Parameters, Tight Security, and Anonymity. Des. Codes Cryptogr. (2021)
https://doi.org/10.1007/s10623-021-00895-3
29. Takayasu, A., Watanabe, Y.: Lattice-Based Revocable Identity-Based Encryption with Bouned Decryption Key Exposure Resistance. In: Pieprzyk, J., Suriadi, S. (eds.) ACISP 2017. LNCS, vol. 10342, pp. 184–204. Springer, Cham (2017)
https://doi.org/10.1007/978-3-319-60055-0_10
30. Wang, C., Fang, J., Li, Y.: An Improved Cloud-Based Revocable Identity-Based Proxy Re-Rncryption Scheme. In: Niu, W., Li, G., Liu, J., et al. (eds.) ATIS 2015. LNCS, vol. 557, pp. 14–26. Springer, Heidelberg (2015)
https://doi.org/10.1007/978-3-662-48683-2_2