Revocable IBE with En-DKER from Lattices: A Novel Approach for Lattice Basis Delegation

Qi Wang[1], Haodong Huang[1], and Juyan Li[1(✉)]

College of Data Science and Technology, Heilongjiang University, Harbin, China
wangamyqi@gmail.com, lijuyan@hlju.edu.cn

Abstract. In public key encryption (PKE), anonymity is essential to ensure privacy by preventing the ciphertext from revealing the recipient's identity. However, the literature has addressed the anonymity of PKE under different attack scenarios to a limited extent. Benhamouda et al. (TCC 2020) introduced the first formal definition of anonymity for PKE under corruption, and Huang et al. (ASIACRYPT 2022) made further extensions and provided a generic framework.

In this paper, we introduce a new security notion named enhanced decryption key exposure resistance (En-DKER) for revocable identity-based encryption (RIBE). This notion ensures that the exposure of decryption keys within any time period will not compromise the confidentiality and anonymity of ciphertexts encrypted during different periods. Meanwhile, we construct the first RIBE scheme with En-DKER and prove its security under the learning with errors (LWE) assumption. Our scheme offers several advantages. Firstly, the periodic workload of the key generation center (KGC) in our scheme is nearly zero. Secondly, the encryptor does not need to handle real-time revocation information of users within the system. Thirdly, the size of user secret keys remains constant in multi-bit encryption.

Additionally, we present a novel approach to delegate a lattice basis. Diverging from the work of Cash et al. (J CRYPTOL 2012), our approach allows for the outsourcing of subsequent sampling operations to an untrusted server. Leveraging this approach, our scheme significantly reduces the periodic workload for users to generate decryption keys. Finally, we efficiently implemented our scheme using the number theory library (NTL) and multi-threaded parallel program. The experimental results confirm the advantages of our scheme.

Keywords: Revocable identity-based encryption · Anonymity · Decryption key exposure · Lattice-based cryptography · Lattice basis delegation.

1 Introduction

Identity-based encryption (IBE) is an advanced form of public key encryption (PKE) that eliminates the need for certificates by allowing any string to serve as a user's public key. This simplifies the traditional PKE process, but poses a

challenge when revoking malicious users without certificate invalidation mechanism.

Boneh and Franklin [9] proposed a solution in which the key generation center (KGC) periodically generates and broadcasts keys for all non-revoked users. However, their scheme incurs a periodic workload of $O(N - r)$ for the KGC, which can become the system's bottleneck as the number of users grows, where N is the maximum number of users and r is the number of revoked users. Boldyreva et al. [8] proposed an indirect revocation model, that employs a binary tree structure and subset-cover framework, to reduce the periodic workload of the KGC to $O(r \log(N/r))$.

In order to ensure the comprehensive utilization of the revocable identity-based encryption (RIBE) scheme, it is imperative to consider additional attack scenarios and privacy requirements. Key exposure happens frequently due to external attacks or user errors. Seo and Emura [24] introduced an important security notion called decryption key exposure resistance (DKER), which requires that the exposure of decryption keys for any time period cannot compromise the confidentiality of ciphertexts that are encrypted for different time periods within RIBE schemes. It is conceivable that by re-randomize the decryption keys at each time period, the algorithm can satisfy the DKER property, and there are some RIBE schemes with DKER based on number theoretical assumptions, such as bilinear maps and multilinear maps [13, 14, 27].

However, the algebraic structure of lattices, which is believed to be resistant against quantum attacks, has traditionally been considered unsuitable for the key re-randomization property. This is because if a user generates a new decryption key that satisfies the correctness without knowledge of the trapdoor, he can also solve the small integer solution (SIS) problem. Fortunately, Cash et al. [11] introduced a new lattice-based cryptographic structure called bonsai tree, which can be used to achieve lattice basis delegation. In other words, this structure allows for the straightforward extension of any short basis from a parent lattice to a short basis of any higher-dimensional child lattice, enabling key re-randomization in lattice-based systems. Furthermore, Cash et al. utilized the bonsai tree structure to construct the first lattice-based hierarchical identity-based encryption (HIBE) scheme and an efficient 'hash-and-sign' signature scheme in the standard model.

Furthermore, anonymity is a vital privacy requirement in IBE schemes [6], requiring the ciphertext does not reveal the recipient's identity.

Open Problem: *If decryption key exposure for any time period, is it possible to construct an RIBE scheme that ensures the confidentiality and anonymity of ciphertexts encrypted for different time periods?*

However, the anonymity of PKE under different attack scenarios is less studied in the literature. Recently, Benhamouda et al. [7] introduced the first formal definition of anonymity for PKE under corruption. Then, Huang et al. [16] provided a generic framework of the anonymous PKE scheme under corruption. Moreover, Boyen and Waters [10] mentioned that anonymity appears unattainable when re-randomization elements are included in the public parameters. To

the best of our knowledge, there is currently no RIBE scheme that can address the aforementioned problem.

1.1 Motivation and Related Works

Revocation Models. Following the work of Boldyreva et al. [8], Attrapadung and Imai [5] introduced a direct revocation model that eliminates the need for periodic key updates by both the KGC and users. Under this model, data owners can manage the revocation list and generate ciphertext that can only be decrypted by non-revoked users within specific scenarios. However, aside from its limited applicability, this model is restricted to fine-grained revocable encryption schemes, such as revocable attribute encryption (RABE) [19] and revocable predicate encryption (RPE) [18]. For a single recipient, the data owner can verify the non-revocation status of the recipient and share data using IBE schemes without needing RIBE schemes. In 2015, Qin et al. [22] proposed a server-aided revocation model in which almost all user workloads are delegated to an untrusted server. However, the periodic workload of the KGC is still remains logarithmic.

Lattice-Based RIBE with DKER. Chen et al. [12] constructed the first lattice-based RIBE scheme. However, this scheme does not satisfy the DKER property. Until 2019, Katsumata et al. [17] introduced an approach to achieve the partial key re-randomization property from lattices and constructed the first lattice-based RIBE scheme with DKER. Their scheme has a two-level structure, where the first level incorporates Chen et al.'s lattice-based RIBE scheme [12] to ensure that only non-revoked users can decrypt the ciphertext, while the second level relies on any lattice-based HIBE scheme [1,11] to meet the DKER property. By following the idea, Wang et al. [26] constructed a more efficient scheme, and Zhang et al [28] proposed a lattice-based server-aided RIBE with DKER. However, if the decryption key for any time period is exposed, this two-level structure fails to ensure the anonymity of ciphertexts encrypted during different time periods. Takayasu and Watanabe [25] explained this point in detail and addressed a weak version of the aforementioned open problem. They constructed an anonymity RIBE scheme with bounded decryption key exposure resistance (B-DKER) which means the security of RIBE schemes can be guaranteed in the case of a-priori bounded number of decryption keys exposure.

1.2 Our Contributions

This paper presents five significant contributions.

A stronger security notion named En-DKER. Under the assumption that the decryption key will be exposed in any time period, we propose a stronger security notion named enhanced decryption key exposure resistance (En-DKER). For details, see Sect. 3.1.

The security definition of the RIBE scheme with En-DKER. Although some RIBE schemes have anonymous analysis, they lack normative security proof. We propose a security definition for the notion of En-DKER, which can prove the anonymity as well as the security. See Sect. 3.2 for details.

A Novel Approach for Lattice Basis Delegation. This paper presents a novel approach to achieving the lattice basis delegation, which enables the outsourcing of subsequent sampling operations to an untrusted server. For details, see Sect. 4.1.

An RIBE scheme with En-DKER. We construct the first RIBE scheme with En-DKER, which is suitable for multi-bit encryption and scenarios where the KGC has a high computational workload. In addition, we outsource the majority of user's workload to an untrusted server. Finally, we prove the security of our scheme under the LWE assumption. For details, see Sect. 4.

Implementation and evaluation. Our scheme is efficiently implemented through the number theory library (NTL) and multi-threaded parallel programming. The experimental results validate the benefits of our revocation model and scheme. See Sect. 5.1 for details.

2 Preliminaries

2.1 Notations

Throughout this paper, we denote λ as the security parameter. For two distributions $\mathcal{D}$ and $\mathcal{D}'$, the statistical distance between $\mathcal{D}$ and $\mathcal{D}'$ is defined as $\mathsf{SD}(\mathcal{D}, \mathcal{D}')$. A family of distributions $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{D}' = \{\mathcal{D}'_\lambda\}_{\lambda\in\mathbb{N}}$ are said to be statistically indistinguishable if there is a negligible function $\mathsf{negl}(\cdot)$ such that $\mathsf{SD}(\mathcal{D}_\lambda, \mathcal{D}'_\lambda) \leq \mathsf{negl}(\lambda)$ for all $\lambda \in \mathbb{N}$, where $\mathsf{negl}(\cdot)$ represents a function that for every constant $c > 0$ there exists an integer N_c satisfying $\mathsf{negl}(\lambda) \leq \lambda^{-c}$ for all $\lambda > N_c$. Let PPT denote probabilistic polynomial time.

If n is a positive integer, we let $[n] = \{1, \ldots, n\}$. For a vector $\boldsymbol{x} \in \mathbb{Z}_n$, $||\boldsymbol{x}||$ denotes the standard Euclidean norm of $\boldsymbol{x}$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, denote $\widetilde{\boldsymbol{A}}$ as the Gram-Schmidt orthogonalization of matrix $\boldsymbol{A}$ and denote $||\boldsymbol{A}||$ as the Euclidean norm of the longest column in $\boldsymbol{A}$.

Smudging The given lemma, originally established in [4], asserts that adding large noise can "smudges out" any small values.

Definition 1 (B-Bounded). *For a family of distributions $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ over the integers and a bound $\mathcal{B} = \mathcal{B}(\lambda) > 0$, if for every $\lambda \in \mathbb{N}$ it holds that $Pr_{x\leftarrow\mathcal{D}_\lambda}[|x| \leq \mathcal{B}(\lambda)] = 1$, we say that $\mathcal{D}$ is $\mathcal{B}$-bounded.*

Lemma 1 (Smudging Lemma). *Let B_1, B_2 be two polynomials over the integers, and let $\mathcal{D} = \{\mathcal{D}_\lambda\}_\lambda$ be any B_1-bounded distribution family. Let $\mathcal{U} = \{\mathcal{U}_\lambda\}_\lambda$ be the uniform distribution over $[-B_2(\lambda), B_2(\lambda)]$. The family of distributions $\mathcal{D} + \mathcal{U}$ and $\mathcal{U}$ are statistically indistinguishable if there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ it holds that $B_1(\lambda)/B_2(\lambda) \leq \mathsf{negl}(\lambda)$.*

Leftover Hash Lemma Here, we recall the leftover hash lemma from [1].

Lemma 2. *Suppose that $m > (n+1)\log q + \omega(\log n)$, and $k = k(n)$ be some polynomial in n. Then, the distribution $(\boldsymbol{A}, \boldsymbol{AR})$ is statistically indistinguishable to the distribution $(\boldsymbol{A}, \boldsymbol{B})$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are uniformly matrices in $\mathbb{Z}_q^{n\times m}$ and $\mathbb{Z}_q^{n\times k}$, and $\boldsymbol{R}$ is a uniformly matrix in $\{-1,1\}^{n\times k}$. Simultaneously, there exists a constant c such that $\mathsf{Pr}[||\boldsymbol{R}|| > c\sqrt{m+k}] \leq \mathsf{negl}(m)$.*

Full-Rank Different Map We need this tool to encode identities and time periods as matrices in $\mathbb{Z}_q^{n\times n}$.

Definition 2. *A function $\mathsf{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is a full-rank different map if the matrix $\mathsf{H}(\boldsymbol{u}) - \mathsf{H}(\boldsymbol{v}) \in \mathbb{Z}_q^{n\times n}$ is full rank, for all distinct $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}_q^n$, and H is computable in $\mathcal{O}(n\log q)$.*

2.2 Background on Lattices

Lattice. An m-dimensional lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$. Let $\mathcal{L}_q^{\perp}(\boldsymbol{A})$ denote the q-ary lattice $\{\boldsymbol{x} \in \mathbb{Z}^m \mid \boldsymbol{A}\boldsymbol{x}^\top = \boldsymbol{0}^\top \bmod q\}$, where n, m, q are positive integers and $\boldsymbol{A}$ is a matrix in $\mathbb{Z}_q^{n\times m}$. For any $\boldsymbol{u}$ in $\mathbb{Z}_q^n$, let $\mathcal{L}_q^{\boldsymbol{u}}(\boldsymbol{A})$ denote the coset $\{\boldsymbol{x} \in \mathbb{Z}^m \mid \boldsymbol{A}\boldsymbol{x}^\top = \boldsymbol{u}^\top \bmod q\}$.

Discrete Gaussians. For any parameter $\sigma > 0$, the discrete Gaussian distribution $\rho_{\mathcal{L},\sigma}(\boldsymbol{x}) = \rho_\sigma(\boldsymbol{x})/\rho_\sigma(\mathcal{L})$, where $\rho_\sigma(\boldsymbol{x}) = \exp(-\pi||x||^2/\sigma^2)$ and $\rho_\sigma(\mathcal{L}) = \sum_{\boldsymbol{x}\in\mathcal{L}} \rho_\sigma(\boldsymbol{x})$. The following lemmas are important properties of discrete Gaussian [15].

Lemma 3. *Let n, m, q be positive integers with $m > n$, $q > 2$, and $\boldsymbol{A}$ be a matrix in $\mathbb{Z}_q^{n\times m}$. Then, there is a negligible function $\mathsf{negl}(\cdot)$ such that $\mathsf{Pr}[||\boldsymbol{x}|| > \sigma\sqrt{m} : \boldsymbol{x} \leftarrow \mathcal{D}_{\mathcal{L}_q^{\perp}(\boldsymbol{A}),\sigma}] \leq \mathsf{negl}(n)$, when $\sigma = \tilde{\Omega}(n)$.*

Lemma 4. *Let n, m, q be positive integers with $m > 2n\log q$. Then, for $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$, the distribution of $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{e} \bmod q$ is statistically close to the uniform distribution over $\mathbb{Z}_q^n$.*

Sampling Algorithms. We review some sampling algorithms from [2,3,20].

Lemma 5. *Let $n \geq 1$, $m \geq 2n\lceil\log q\rceil$, $q \geq 2$, we have the following polynomial time algorithms:*

- $\mathsf{TrapGen}(1^n, 1^m, q) \to (\boldsymbol{A}, \boldsymbol{T_A})$*: On input n, m, q, output a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and its trapdoor $\boldsymbol{T_A} \in \mathbb{Z}^{m\times m}$, satisfying $||\boldsymbol{T_A}|| \leq O(n\log q)$.*
- $\mathsf{SamplePre}(\boldsymbol{A}, \boldsymbol{T_A}, \sigma, \boldsymbol{u}) \to \boldsymbol{s}$*: On input a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and its trapdoor $\boldsymbol{T_A}$, a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, and a parameter $\sigma \geq ||\widetilde{\boldsymbol{T_A}}|| \cdot \omega(\sqrt{\log m})$, output a vector $\boldsymbol{s} \in \mathbb{Z}_q^m$, satisfying $\boldsymbol{A}\cdot\boldsymbol{s}^\top = \boldsymbol{u}^\top$ and $||\boldsymbol{s}|| \leq \sqrt{m}\sigma$.*

– $\mathsf{SampleLeft}(\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{T_A}, \sigma, \boldsymbol{u}) \to \boldsymbol{s}$: *On input a matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ *and its trapdoor* $\boldsymbol{T_A}$*, a matrix* $\boldsymbol{M} \in \mathbb{Z}_q^{n \times m_0}$*, a vector* $\boldsymbol{u} \in \mathbb{Z}_q^n$*, and a parameter* $\sigma \geq ||\widetilde{\boldsymbol{T_A}}|| \cdot \omega(\sqrt{\log(m + m_0)})$*, output a vector* $\boldsymbol{s} \in \mathbb{Z}_q^{m+m_0}$ *distributed statistically close to* $\mathcal{D}_{\mathcal{L}_q^{\boldsymbol{u}}([\boldsymbol{A}|\boldsymbol{M}]),\sigma}$.

– *There is a gadget matrix* $\boldsymbol{G}$*, which is a full rank matrix in* $\mathbb{Z}_q^{n \times m}$ *and has a publicly known trapdoor* $\boldsymbol{T_G}$ *with* $||\widetilde{\boldsymbol{T_G}}|| \leq \sqrt{5}$.

– $\mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{G}, \boldsymbol{R}, \boldsymbol{T_G}, \sigma, \boldsymbol{u}) \to \boldsymbol{s}$: *On input a matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$*, the gadget matrix* $\boldsymbol{G}$ *and its trapdoor* $\boldsymbol{T_G}$*, a uniform random matrix* $\boldsymbol{R} \leftarrow \{-1, 1\}^{m \times m}$*, a vector* $\boldsymbol{u} \in \mathbb{Z}_q^n$*, and a parameter* $\sigma \geq ||\widetilde{\boldsymbol{T_G}}|| \cdot \sqrt{m} \cdot \omega(\sqrt{\log m})$*, output a vector* $\boldsymbol{s} \in \mathbb{Z}_q^{2m}$ *distributed statistically close to* $\mathcal{D}_{\mathcal{L}_q^{\boldsymbol{u}}([\boldsymbol{A}|\boldsymbol{AR}+\boldsymbol{G}]),\sigma}$.

LWE Assumption. Our RIBE scheme is based on the learning with errors (LWE) assumption.

Assumption 1 (Learning with Errors [23]). *Let* n*,* m*,* q *be positive integers, and a parameter* $\sigma \in \mathbb{R}$*, for any PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *that satisfies* $|\mathsf{Pr}[\mathcal{A}(\boldsymbol{\alpha}, \boldsymbol{s}^\top \boldsymbol{\alpha} + e) = 1] - \mathsf{Pr}[\mathcal{A}(\boldsymbol{\alpha}, \gamma) = 1]| \leq \mathsf{negl}(\lambda)$*, where* $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_q^n$*,* $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$*,* $\gamma \leftarrow \mathbb{Z}_q$*, and* $e \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}$.

2.3 The Complete Subtree Method

The complete subtree (CS) method, proposed by Naor et al. [21], effectively improves the efficiency of the revocation schemes. In this method, the system will build a complete binary tree BT. For a non-leaf node $\theta \in \mathsf{BT}$, θ_l and θ_r denote the left and right child node of θ, and η denote the leaf node in BT. $\mathsf{Path}(\eta)$ denote the set of nodes on the path from η to the root. Inputting the revocation list RL_t on the time period t, then the $\mathsf{KUNodes}$ algorithm proceeds as follows: sets two empty sets X and Y; adds $\mathsf{Path}(\eta)$ to X, for each $\eta \in \mathsf{RL}_t$; for each $\theta \in X$, adds θ_l to Y if $\theta_l \notin X$, adds θ_r to Y if $\theta_r \notin X$; if Y is still the empty set, then adds root to Y; finally, outputs Y which is the smallest nodes subset of non-revoked users on the time period t.

3 Formal Definition for RIBE with En-DKER

3.1 Enhanced Decryption Key Exposure Resistance

Definition 3 (En-DKER). *The exposure of users' decryption keys for any time period does not compromise the anonymity and confidentiality of ciphertexts that are encrypted for different time periods.*

It should be noted that En-DKER is different from achieving both DKER and anonymity since current anonymous IBE schemes are constructed under the assumption that the user's decryption keys will not be exposed. In other words, RIBE with En-DKER scheme cannot be constructed by simply combining the RIBE with DKER scheme with an anonymous IBE scheme. Therefore, it is necessary to define a new security notion to avoid confusion for readers.

3.2 Syntax of RIBE with En-DKER

Definition of RIBE. Our revocation model is different from the previous ones. Specifically, in the direct revocation model proposed by Attrapadung and Imai [5], the encryptor uses the revocation list RL_t to generate the set $\mathsf{KUNodes}(\mathsf{RL}_t)$ which represents the smallest nodes subset of non-revoked users on time period t. What is interesting is that the set $\mathsf{KUNodes}(\mathsf{RL}_t)$ does not reveal any information about the revocation list RL_t since the adversary is unable to determine which user corresponds to each leaf node. Therefore, in our model, the KGC periodically generates and broadcasts the set $\mathsf{KUNodes}(\mathsf{RL}_t)$, thereby eliminating the encryptor's need to handle any revocation list information and making our model free from specific scenarios. Moreover, our model inherits the benefits of the direct revocation model, and the periodic workload of the KGC is nearly zero.

Our RIBE scheme consists of the six algorithms (Setup, GenSK, $\mathsf{NodesUp}$, GenDK, Enc, Dec) with associated message space $\mathcal{M}$, identity space $\mathcal{ID}$, and time period space $\mathcal{T}$. The KGC maintains a revocation list RL which is dynamically updated following the time period t.

- $\mathbf{Setup}(\lambda, N)$: This algorithm is run by the KGC. Input a security parameter λ and a maximal number N of users, output public parameters PP and a master secret key MSK.
- $\mathbf{GenSK}(\mathsf{PP}, \mathsf{MSK}, \mathsf{ID})$: This algorithm is run by the KGC. Input the public parameters PP, the master secret key MSK, and an identity $\mathsf{ID} \in \mathcal{ID}$, output a secret key $\mathsf{SK}_{\mathsf{ID}}$ for the user with the identity ID.
- $\mathbf{NodesUp}(\mathsf{BT}, \mathsf{RL}_t)$: This algorithm is run by the KGC. Input the binary tree BT and the revocation list RL_t, the KGC generates and broadcasts a node set $\mathsf{KUNodes}(\mathsf{RL}_t)$ for the time period t.
- $\mathbf{GenDK}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KUNodes}(\mathsf{RL}_t))$: This algorithm is run by the receiver. Input the public parameters PP, the secret key $\mathsf{SK}_{\mathsf{ID}}$, and the set $\mathsf{KUNodes}(\mathsf{RL}_t)$, output a decryption key $\mathsf{DK}_{\mathsf{ID},t}$.
- $\mathbf{Enc}(\mathsf{PP}, \mathsf{ID}, t, \mathsf{KUNodes}(\mathsf{RL}_t), msg)$: This algorithm is run by the sender. Input the public parameters PP, an identity $\mathsf{ID} \in \mathcal{ID}$, a time period $t \in \mathcal{T}$, the set $\mathsf{KUNodes}(\mathsf{RL}_t)$, and message msg, output a ciphertext $\mathsf{CT}_{\mathsf{ID},t}$.
- $\mathbf{Dec}(\mathsf{CT}_{\mathsf{ID},t}, \mathsf{DK}_{\mathsf{ID},t})$: This algorithm is run by the receiver. Input the ciphertext $\mathsf{CT}_{\mathsf{ID},t}$ and the decryption key $\mathsf{DK}_{\mathsf{ID},t}$, output message $msg' \in \mathcal{M}$.

Correctness. An RIBE scheme is correct if for all $\lambda \in \mathbb{N}$, $N \in \mathbb{N}$, $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(\lambda, l, N)$, $msg \in \mathcal{M}$, $\mathsf{ID} \in \mathcal{ID}$, $t \in \mathcal{T}$ and revocation lists RL it holds that

$$\Pr\left[msg' = msg \;\middle|\; \begin{array}{c} \mathsf{SK}_{\mathsf{ID}} \leftarrow \mathbf{GenSK}(\mathsf{PP}, \mathsf{MSK}, \mathsf{ID}) \\ \mathsf{KUNodes}(\mathsf{RL}_t) \leftarrow \mathbf{NodesUp}(\mathsf{BT}, \mathsf{RL}_t) \\ \mathsf{DK}_{\mathsf{ID},t} \leftarrow \mathbf{GenDK}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KUNodes}(\mathsf{RL}_t)) \\ \mathsf{CT}_{\mathsf{ID},t} \leftarrow \mathbf{Enc}(\mathsf{PP}, \mathsf{ID}, t, \mathsf{KUNodes}(\mathsf{RL}_t), msg) \\ msg' \leftarrow \mathbf{Dec}(\mathsf{CT}_{\mathsf{ID},t}, \mathsf{DK}_{\mathsf{ID},t}) \end{array} \right] = 1.$$

Security. Now, we give a formal security definition for RIBE with En-DKER by the game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$. Different from the security definition of RIBE with DKER, we replace the challenge identity ID with $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$. When $\mathcal{C}$ randomly chooses a bit b, the challenge plaintext $msg^{(b)}$ will be encrypted with the identity $\mathsf{ID}^{(b)}$. Assuming the scheme does not satisfy anonymity, the adversary can distinguish between $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$, then get the value of challenge bit b and win the game. So in this setting, our security definition can verify the anonymity while proving the security of the RIBE schemes.

In addition, since the revocation list RL is dynamically updated following the time period t, so we set a global variable $t_{cu} \in \mathcal{T}$, whose initial value is 1, to assist in generating the decryption key $\mathsf{DK}_{\mathsf{ID},t}$ of any time period queried by $\mathcal{A}$.

Initial: $\mathcal{A}$ sets the challenge identities $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$, the challenge time period t^*, and the challenge node set $\mathsf{KUNodes}(\mathsf{RL}_{t^*})^*$.

Setup Phase: $\mathcal{C}$ runs the Setup algorithm and gives the public parameters PP to $\mathcal{A}$.

Query Phase: $\mathcal{A}$ adaptively makes a polynomial number of the following queries to $\mathcal{C}$:

1. $\mathcal{A}$ sets $\mathcal{Q}_0 = \{\mathsf{ID}\}$ for the establishment of the binary tree BT. $\mathcal{C}$ randomly picks an unassigned leaf node η_{ID} for ID.[1] At the end of the quiry, $\mathcal{C}$ obtains $\mathsf{RL}^*_{t^*}$ based on $\mathsf{KUNodes}(\mathsf{RL}_{t^*})^*$ and BT, and sends it to $\mathcal{A}$.
2. $\mathcal{A}$ sets $\mathcal{Q}_1 = \{\mathsf{ID}\}$ for the secret key queries, subject to the restriction: $\mathsf{ID} \in \mathcal{Q}_0$; if $\mathsf{ID} = \mathsf{ID}^{(0)}$ or $\mathsf{ID}^{(1)}$, $\mathsf{ID} \in \mathsf{RL}^*_{t^*}$. $\mathcal{C}$ replies with the corresponding secret key $\mathsf{SK}_{\mathsf{ID}} \leftarrow \mathsf{GenSK}(\mathsf{PP}, \mathsf{MSK}, \mathsf{ID})$.
3. Let $t_{cu} = 1$, and loop through the following steps:
   (a) $\mathcal{A}$ sets $\mathcal{Q}_2 = \{(\mathsf{ID}, t_{cu})\}$ for the decryption key queries, subject to the restriction: $\mathsf{ID} \in \mathcal{Q}_0$; $\mathsf{ID} \notin \mathsf{RL}_{t_{cu}}$; if $t_{cu} = t^*$, $\mathsf{ID} \neq \mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$. $\mathcal{C}$ replies with the decryption key $\mathsf{DK}_{\mathsf{ID},t} \leftarrow \mathsf{GenDK}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KUNodes}(\mathsf{RL}_t))$.
   (b) $\mathcal{A}$ sets $\mathcal{Q}_3 = \{(\mathsf{ID}, t_{cu})\}$ for revocation queries, subject to the restriction: $\mathsf{ID} \in \mathcal{Q}_0$; $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$ are either queried at the same time period t or neither,[2]; $\mathsf{RL}_{t^*} = \mathsf{RL}^*_{t^*}$. $\mathcal{C}$ adds ID to the revocation list RL, and updates $\mathsf{RL}_{t_{cu}+1} = \mathsf{RL}$. Then, $\mathcal{C}$ sent $\mathsf{KUNodes}(\mathsf{RL}_{t_{cu}+1})$ to $\mathcal{A}$.
   (c) $t_{cu} = t_{cu} + 1$.

Challenge Phase: $\mathcal{A}$ outputs the challenge plaintexts $msg^{(0)}$ and $msg^{(1)}$. Then $\mathcal{C}$ chooses a random bit $b \leftarrow \{0, 1\}$ and replies with the corresponding ciphertext $\mathsf{CT}_{\mathsf{ID}^{(b)},t^*} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ID}^{(b)}, t^*, \{msg_i^{(b)}\}_{i\in[l]})$.

Guess: $\mathcal{A}$ outputs a guess b' of b.

Definition 4. *An RIBE with En-DKER scheme is selectively secure if the advantage* $\mathsf{Adv}^{\mathsf{SEL\text{-}En\text{-}CPA}}_{\mathsf{RIBE},\mathcal{A}}(\lambda)$ *is at most negligible for any PPT adversaries* $\mathcal{A}$*, where* $\mathsf{Adv}^{\mathsf{SEL\text{-}En\text{-}CPA}}_{\mathsf{RIBE},\mathcal{A}}(\lambda) = |\mathsf{Pr}[b = b'] - 1/2|$.

[1] This step moves from the algorithm GenSK to the Query Phase.

[2] If the two challenge identities are revoked at different time periods, the adversary can distinguish them in the subsequent key queries phase.

Remark 1. According to the challenge identities $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$, and the challenge time period t^*, it needs to be divided into two cases:

– If $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$ are revoked before t^*, adversary $\mathcal{A}$ can perform the secret key queries and decryption key queries according to the corresponding restrictions.

– If $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$ have not been revoked before t^*, $\mathcal{A}$ can perform decryption key queries according to the corresponding restrictions. It is important to note that the RIBE without En-DKER schemes cannot support queries in this case.

4 Revocable IBE with En-DKER from Lattices

In this section, we present our proposed lattice-based RIBE scheme with En-DKER. We begin by introducing our approach for lattice basis delegation in Sect. 4.1. In Sect. 4.2, we explain the core techniques and the main idea behind our scheme. Finally, we present our scheme in Sect. 4.3 and prove the security in Sect. 4.4.

4.1 Lattice Basis Delegation

Lattice basis delegation allows for the extension of any short basis $\boldsymbol{T_A}$ from a parent lattice $\boldsymbol{A}$ to a trapdoor of any higher-dimensional child lattice $[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]$, where the child's trapdoor cannot leak any information of $\boldsymbol{T_A}$. Then, the child can use this trapdoor to sample decryption keys. Our novel approach can outsource this sampling calculation to an untrusted server. The details are as follows.

First, the KGC runs the algorithm $\mathsf{TrapGen}$ to generate two pairs of matrices with trapdoors $(\boldsymbol{A}, \boldsymbol{T_A})$ and $(\bar{\boldsymbol{A}}, \boldsymbol{T}_{\bar{\boldsymbol{A}}})$, where $\{\boldsymbol{A}, \bar{\boldsymbol{A}}, \boldsymbol{T}_{\bar{\boldsymbol{A}}}\}$ is the public parameters PP and $\boldsymbol{T_A}$ is the master secret key MSK. Then, by utilizing the $\mathsf{SampleLeft}$ algorithm and the master secret key MSK, the KGC generates $\boldsymbol{sk}_{\mathsf{ID}}$, satisfying $[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{sk}_{\mathsf{ID}} = \bar{\boldsymbol{A}}$. Meanwhile, $\boldsymbol{sk}_{\mathsf{ID}}$ can serve as the trapdoor for the child, because for any vector $\boldsymbol{x} \in \mathbb{Z}_q^n$, the user can also calculate a bounded small key $\boldsymbol{k}$ by using $\boldsymbol{sk}_{\mathsf{ID}}$, satisfying $[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{k} = \boldsymbol{x}$. The difference is that the majority of the workload to generate $\boldsymbol{k}$ can be outsourced to an untrusted server. Specifically, by utilizing the $\mathsf{SampleLeft}$ algorithm and the public trapdoor $\boldsymbol{T}_{\bar{\boldsymbol{A}}}$, the server generates $\boldsymbol{k}'$ and sends it to the user, satisfying $\bar{\boldsymbol{A}}\boldsymbol{k}' = \boldsymbol{x}$. Then, the user only needs to calculate $\boldsymbol{sk}_{\mathsf{ID}}\boldsymbol{k}'$ as the key $\boldsymbol{k}$.

However, $\boldsymbol{sk}_{\mathsf{ID}}\boldsymbol{k}'$ is only a bounded small key. To make the key $\boldsymbol{k}$ satisfy the re-randomization property, we introduce an important tool called smudging lemma [4]. Specifically, the user first uniformly select an random vector $\boldsymbol{sk'}$ in a relatively large distribution, and set $\boldsymbol{x'} = \boldsymbol{x} - [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{sk'}$. Subsequently, by employing the sampling outsourcing approach, the server can generate the key $\boldsymbol{k}'$, satisfying $\bar{\boldsymbol{A}}\boldsymbol{k}' = \boldsymbol{x}'$. The user can obtain the key $\boldsymbol{k}$ by adding $\boldsymbol{sk}'$ and $\boldsymbol{sk}_{\mathsf{ID}}\boldsymbol{k'}$ in a component-wise fashion, satisfying $[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{k} = \boldsymbol{x}$. Smudging lemma can guarantee the randomness of the decryption key.

Correctness. Now, we analyze the correctness of our approach.

$$\begin{aligned}[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\,\boldsymbol{k} &= [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{s}\boldsymbol{k}' + [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{s}\boldsymbol{k}_{\mathsf{ID}}\boldsymbol{k}' \\ &= [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{s}\boldsymbol{k}' + \bar{\boldsymbol{A}}\boldsymbol{k}' \\ &= [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{s}\boldsymbol{k}' + \boldsymbol{x} - [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{s}\boldsymbol{k}' = \boldsymbol{x}.\end{aligned}$$

4.2 Technique Review

Katsumata et al. [17] constructed the first lattice-based RIBE scheme with DKER, which utilized Cash et al.'s lattice basis delegation [11] to achieve partial re-randomization of the decryption keys and attain the DKER property. However, their scheme suffered from a limitation in which the exposure of another non-randomizable partial decryption key compromised the anonymity of the ciphertext. To address this limitation, we employ the lattice basis delegation approach (as described in Sect.4.1) to achieve full re-randomization of our decryption keys within each time period. Furthermore, unlike key re-randomization based on number-theoretical assumptions, lattice basis delegation does not necessitate any modifications to the public parameters or ciphertexts. Based on these two points, we construct the first RIBE scheme with En-DKER.

Multi-Bit Encryption. Agrawal et al. [1] proposed an approach for multi-bit encryption, in which encrypts l bits message using a single random vector $\boldsymbol{s} \in \mathbb{Z}_q^n$. Specifically, they set l vectors $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_l)$ from $\mathbb{Z}_q^n$ into the public parameters PP, as opposed to the basic scheme which utilizes only a single vector $\boldsymbol{u}$. Message bit number i is encrypted using the vector $\boldsymbol{u}_i$. Our scheme follows this approach to achieve multi-bit encryption.

However, in current lattice-based RIBE schemes, changing the vector $\boldsymbol{u}$ from one column to l column results in the size of user secret keys, update keys, and decryption keys growing by a factor of l. The workload for the KGC and users also increases by a factor of l. Fortunately, in our scheme, the size of user secret keys remains constant, periodic workload of the KGC remains nearly zero, and the majority of the workload for generating decryption keys is outsourced to the server with the advantages of our lattice basis delegation approach.

4.3 Construction

In our scheme, we set the message space $\mathcal{M} = \{0, 1\}$, the identity space $\mathcal{ID} \subset \mathbb{Z}_q^n \setminus \{\boldsymbol{0}_n\}$, and the time period space $\mathcal{T} \subset \mathbb{Z}_q^n$. For any $B \in \mathbb{N}$, let $\mathcal{U}_B$ denote the uniform distribution on $\mathbb{Z} \cap [-B, B]$. In addition, our system parameters satisfy the following constraints: $m > 2n \log q$ and $\sigma > \sqrt{m} \cdot \omega(\sqrt{m})$ (for sampling); $O(m^{3/2}\mathsf{B}\sigma) < q/4$ (for correctness); $n = O(\lambda)$, $\chi_{\mathsf{LWE}} = \mathcal{D}_{\mathbb{Z},\sigma}$ (for security); $\chi_{\mathsf{big}} = \mathcal{U}_B$, where $B > (m\sigma^2 + 1)2^\lambda$ (for smudging).

Now, we describe our lattice-based RIBE with En-DKER construction.

Setup(λ, l, N): On input a security parameter λ, number of encryption bits l, and maximum number of users N. The specific process is as follows:

1. Choose an LWE modulus q and dimensions n, m.
2. Run the algorithm $\mathsf{TrapGen}(1^n, 1^m, q)$ to generate two pairs of matrices with trapdoors $(\boldsymbol{A}, \boldsymbol{T_A})$ and $(\bar{\boldsymbol{A}}, \boldsymbol{T}_{\bar{\boldsymbol{A}}})$.
3. Select uniformly random matrices $\boldsymbol{B}$, and $\boldsymbol{W}$ in $\mathbb{Z}_q^{n\times m}$, and uniformly random vectors $\{\boldsymbol{u}_i\}_{i\in[l]}$ in $\mathbb{Z}_q^n$.
4. Build a binary tree BT with at least N leaf nodes. For each node $\theta \in \mathsf{BT}$, select a uniformly random matrix $\boldsymbol{D}_\theta$ in $\mathbb{Z}_q^{n\times m}$.
5. Output $\mathsf{PP} = \{\boldsymbol{A}, \bar{\boldsymbol{A}}, \boldsymbol{T}_{\bar{\boldsymbol{A}}}, \boldsymbol{B}, \boldsymbol{W}, \{\boldsymbol{u}_i\}_{i\in[l]}, \{\boldsymbol{D}_\theta\}_{\theta\in\mathsf{BT}}\}$, $\mathsf{MSK} = \{\boldsymbol{T_A}, \mathsf{BT}\}$.

GenSK$(\mathsf{PP}, \mathsf{MSK}, \mathsf{ID})$: On input the public parameters PP, the master secret key MSK, and an identity $\mathsf{ID} \in \mathcal{ID}$. The specific process is as follows:

1. Randomly pick an unassigned leaf node η_{ID} from BT and store ID in it.
2. Set $\boldsymbol{B}_{\mathsf{ID}} = \boldsymbol{B} + \mathsf{H}(\mathsf{ID})\boldsymbol{G}$, where $\mathsf{H}(\cdot)$ is a full-rank different map defined in Definition 2 and $\boldsymbol{G}$ is a gadget matrix defined in Lemma 5.
3. For each $\theta \in \mathsf{Path}(\eta_{\mathsf{ID}})$, generate $\boldsymbol{sk}_{\mathsf{ID},\theta}$ satisfying $[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}|\boldsymbol{D}_\theta]\boldsymbol{sk}_{\mathsf{ID},\theta} = \bar{\boldsymbol{A}}$.
   (a) Set $\boldsymbol{Z}_{\mathsf{ID}} = [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{sk}'_{\mathsf{ID}}$, where $\boldsymbol{sk}'_{\mathsf{ID}}$ is a uniformly random matrix selected in $\chi_{\mathsf{LWE}}^{2m\times m}$.
   (b) Sample $\boldsymbol{sk}''_{\mathsf{ID},\theta} \leftarrow \mathsf{SampleLeft}(\boldsymbol{A}, \boldsymbol{D}_\theta, \boldsymbol{T_A}, \sigma, \bar{\boldsymbol{A}} - \boldsymbol{Z}_{\mathsf{ID}})$.
   (c) Split $\boldsymbol{sk}'_{\mathsf{ID}}$ and $\boldsymbol{sk}''_{\mathsf{ID},\theta}$ into two parts, $\boldsymbol{sk}'_{1,\mathsf{ID}}$, $\boldsymbol{sk}'_{2,\mathsf{ID}}$ and $\boldsymbol{sk}''_{1,\mathsf{ID},\theta}$, $\boldsymbol{sk}''_{2,\mathsf{ID},\theta}$, m rows per part. Then, generate

$$\boldsymbol{sk}_{\mathsf{ID},\theta} = \left[\left(\boldsymbol{sk}'_{1,\mathsf{ID}} + \boldsymbol{sk}''_{1,\mathsf{ID},\theta}\right)^\top \middle| \left(\boldsymbol{sk}'_{2,\mathsf{ID}}\right)^\top \middle| \left(\boldsymbol{sk}''_{2,\mathsf{ID},\theta}\right)^\top\right]^\top \in \mathbb{Z}_q^{3m\times m}.$$

4. Output $\mathsf{SK}_{\mathsf{ID}} = \{\boldsymbol{sk}_{\mathsf{ID},\theta}\}_{\theta\in\mathsf{Path}(\eta_{\mathsf{ID}})}$.

NodesUp$(\mathsf{BT}, \mathsf{RL}_t)$: On input the binary tree BT and the revocation list RL_t, the KGC generates and broadcasts a set $\mathsf{KUNodes}(\mathsf{RL}_t)$ for the time period t.

GenDK$(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KUNodes}(\mathsf{RL}_t))$: On input the public parameters PP, the secret key $\mathsf{SK}_{\mathsf{ID}}$, and the node set $\mathsf{KUNodes}(\mathsf{RL}_t)$. The specific process is as follows:

1. Perform node matching, and let $\theta^* = \mathsf{Path}(\eta_{\mathsf{ID}}) \cap \mathsf{KUNodes}(\mathsf{RL}_t)$. If $\theta^* = \emptyset$, outputs $\perp$. Otherwise, continue the following steps.
2. For $i \in [l]$, generate $\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t}$ satisfying $[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}|\boldsymbol{D}_{\theta^*}|\boldsymbol{W}_t]\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} = \boldsymbol{u}_i$, where $\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} \in \mathbb{Z}_q^{4m}$.
   (a) Set $\boldsymbol{h}_{i,\mathsf{ID},t} = [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}|\boldsymbol{D}_{\theta^*}|\boldsymbol{W}_t]\boldsymbol{k}_{i,t}$ and send to the server, where $\boldsymbol{k}_{i,t}$ is a uniformly random vector selected in χ_{big}^{4m}, $\boldsymbol{W}_t = \boldsymbol{W} + \mathsf{H}(t)\boldsymbol{G}$.
   (b) The server samples $\boldsymbol{k}'_{i,\mathsf{ID},t} \leftarrow \mathsf{SamplePre}(\bar{\boldsymbol{A}}, \boldsymbol{T}_{\bar{\boldsymbol{A}}}, \sigma, \boldsymbol{u}_i - \boldsymbol{h}_{i,\mathsf{ID},t})$ and sends to the user.
   (c) Compute $\boldsymbol{k}''_{i,\mathsf{ID},\theta^*,t} = \boldsymbol{sk}_{\mathsf{ID},\theta^*}\boldsymbol{k}'_{i,\mathsf{ID},t}$, satisfying $[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}|\boldsymbol{D}_{\theta^*}]\boldsymbol{k}''_{i,\mathsf{ID},\theta^*,t} = \boldsymbol{u}_i - \boldsymbol{h}_{i,\mathsf{ID},t}$, where $\boldsymbol{k}''_{i,\mathsf{ID},\theta^*,t} \in \mathbb{Z}_q^{3m}$.
   (d) Split $\boldsymbol{k}_{i,t}$ into four parts, $\boldsymbol{k}_{1,i,t}$, $\boldsymbol{k}_{2,i,t}$, $\boldsymbol{k}_{3,i,t}$, $\boldsymbol{k}_{4,i,t}$, and $\boldsymbol{k}''_{i,\mathsf{ID},\theta^*,t}$ into three parts $\boldsymbol{k}''_{1,i,\mathsf{ID},\theta^*,t}$, $\boldsymbol{k}''_{2,i,\mathsf{ID},\theta^*,t}$, $\boldsymbol{k}''_{3,i,\mathsf{ID},\theta^*,t}$, m rows per part. Then, generate

$$\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} = \left[\begin{pmatrix}\boldsymbol{k}_{1,i,t} + \boldsymbol{k}''_{1,i,\mathsf{ID},\theta^*,t}\\ \boldsymbol{k}_{2,i,t} + \boldsymbol{k}''_{2,i,\mathsf{ID},\theta^*,t}\end{pmatrix}^\top \middle| \begin{pmatrix}\boldsymbol{k}_{3,i,t} + \boldsymbol{k}''_{3,i,\mathsf{ID},\theta^*,t}\\ \boldsymbol{k}_{4,i,t}\end{pmatrix}^\top\right]^\top \in \mathbb{Z}_q^{4m}.$$

3. Output $\mathsf{DK}_{\mathsf{ID},t} = \{\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t}\}_{i\in[l]}$.

Enc$(\mathsf{PP}, \mathsf{ID}, t, \mathsf{KUNodes}(\mathsf{RL}_t), \{msg_i\}_{i\in[l]})$: On input the public parameters PP, an identity $\mathsf{ID} \in \mathcal{ID}$, a time period $t \in \mathcal{T}$, the set $\mathsf{KUNodes}(\mathsf{RL}_t)$, and message $msg_i \in \mathcal{M}$, where $i \in [l]$. The specific process is as follows:

1. Select uniformly random matrices $\boldsymbol{R}$, $\boldsymbol{S}_\theta$, and $\boldsymbol{V}$ in $\{-1,1\}^{m\times m}$, where $\theta \in \mathsf{KUNodes}(\mathsf{RL}_t)$, and a uniformly random vector $\boldsymbol{s}$ in $\mathbb{Z}_q^n$.
2. Choose noise $e_i \leftarrow \chi_{\mathsf{LWE}}$ and a noise vector $\boldsymbol{e}' \leftarrow \chi_{\mathsf{LWE}}^m$, where $i \in [l]$.
3. Set $C_i = \boldsymbol{s}^\top \boldsymbol{u}_i + \lfloor \frac{q}{2} \rfloor \cdot msg_i + e_i$, where $i \in [l]$.
4. Set $\boldsymbol{c}_{\mathsf{ID},\theta,t} = \boldsymbol{s}^\top[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}|\boldsymbol{D}_\theta|\boldsymbol{W}_t] + \boldsymbol{e}'^\top[\boldsymbol{I}_m|\boldsymbol{R}|\boldsymbol{S}_\theta|\boldsymbol{V}]$, where $\boldsymbol{I}_m$ is an identity matrix, $\theta \in \mathsf{KUNodes}(\mathsf{RL}_t)$.
5. Output $\mathsf{CT}_{\mathsf{ID},t} = \{\{C_i\}_{i\in[l]}, \{\boldsymbol{c}_{\mathsf{ID},\theta,t}\}_{\theta\in\mathsf{KUNodes}(\mathsf{RL}_t)}\}$.

Dec$(\mathsf{CT}_{\mathsf{ID},t}, \mathsf{DK}_{\mathsf{ID},t})$: On input the ciphertext $\mathsf{CT}_{\mathsf{ID},t}$ and the decryption key $\mathsf{DK}_{\mathsf{ID},t}$. The specific process is as follows:

1. Compute $C_i' = C_i - \boldsymbol{c}_{\mathsf{ID},\theta^*,t}\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t}$, where $i \in [l]$.
2. For each $i \in [l]$, output $msg_i = 1$ if $|C_i' - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$, otherwise $msg_i = 0$.

Correctness. Now, we analyze the correctness of our scheme,

$$\begin{aligned} C_i' &= C_i - \boldsymbol{c}_{\mathsf{ID},\theta^*,t}\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} \\ &= \boldsymbol{s}^\top \boldsymbol{u}_i + \left\lfloor \frac{q}{2} \right\rfloor \cdot msg_i - \boldsymbol{s}^\top[\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}|\boldsymbol{D}_{\theta^*}|\boldsymbol{W}_t]\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} + \mathsf{noise}_i \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot msg_i + \mathsf{noise}_i, \end{aligned}$$

for each $i \in [l]$, where

$$\begin{aligned} \mathsf{noise}_i &= e_i - \boldsymbol{e}'^\top[\boldsymbol{I}_m|\boldsymbol{R}|\boldsymbol{S}_{\theta^*}|\boldsymbol{V}]\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} \\ &= e_i - \boldsymbol{e}'^\top[\boldsymbol{I}_m|\boldsymbol{R}|\boldsymbol{S}_{\theta^*}|\boldsymbol{V}] \begin{bmatrix} \boldsymbol{k}_{1,i,t} + \boldsymbol{sk}'_{1,\mathsf{ID}}\boldsymbol{k}'_{i,\mathsf{ID},t} + \boldsymbol{sk}''_{1,\mathsf{ID},\theta^*}\boldsymbol{k}'_{i,\mathsf{ID},t} \\ \boldsymbol{k}_{2,i,t} + \boldsymbol{sk}'_{2,\mathsf{ID}}\boldsymbol{k}'_{i,\mathsf{ID},t} \\ \boldsymbol{k}_{3,i,t} + \boldsymbol{sk}''_{2,\mathsf{ID},\theta^*}\boldsymbol{k}'_{i,\mathsf{ID},t} \\ \boldsymbol{k}_{4,i,t} \end{bmatrix}. \end{aligned}$$

Correctness now follows since noise_i is small and should not affect $\lfloor \frac{q}{2} \rfloor \cdot msg_i$. Moreover, the following inequalities hold except with negligible probability:

– From Lemma 2, we have $||\boldsymbol{R}||$, $||\boldsymbol{S}_{\theta^*}||$, and $||\boldsymbol{V}|| \le O(\sqrt{m})$.
– From Lemma 1, we have $||\boldsymbol{k}_{1,i,t}||$, $||\boldsymbol{k}_{2,i,t}||$, $||\boldsymbol{k}_{3,i,t}||$, and $||\boldsymbol{k}_{4,i,t}|| \le \sqrt{m}\mathsf{B}$.
– From Lemma 5, we have $||\boldsymbol{sk}'_{1,\mathsf{ID}}\boldsymbol{k}'_{i,\mathsf{ID},t}||$, $||\boldsymbol{sk}''_{1,\mathsf{ID},\theta^*}\boldsymbol{k}'_{i,\mathsf{ID},t}||$, $||\boldsymbol{sk}'_{2,\mathsf{ID}}\boldsymbol{k}'_{i,\mathsf{ID},t}||$, and $||\boldsymbol{sk}''_{2,\mathsf{ID},\theta^*}\boldsymbol{k}'_{i,\mathsf{ID},t}|| \le m^{3/2}\sigma$, and $||e_i|| \le \sigma$, $||\boldsymbol{e}'|| \le \sqrt{m}\sigma$.

$$\begin{aligned} ||\mathsf{noise}_i|| &= ||e_i - \boldsymbol{e}'^\top[\boldsymbol{I}_m|\boldsymbol{R}|\boldsymbol{S}_{\theta^*}|\boldsymbol{V}]\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t}|| \\ &\le ||e_i|| + ||\boldsymbol{e}'^\top|| \cdot ||[\boldsymbol{I}_m|\boldsymbol{R}|\boldsymbol{S}_{\theta^*}|\boldsymbol{V}]\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t}|| \\ &\le \sigma + (\sqrt{m}\sigma)[(2m^{3/2}\sigma + \sqrt{m}\mathsf{B}) + (2m^{3/2}\sigma + 3\sqrt{m}\mathsf{B})O(\sqrt{m})] \\ &\le O(m^{3/2}\mathsf{B}\sigma) < q/4, \end{aligned}$$

and we can get msg_i by judging $|C_i' - \lfloor \frac{q}{2} \rfloor| = |msg_i \cdot \lfloor \frac{q}{2} \rfloor + \mathsf{noise}_i - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$.

4.4 Security Analysis

Theorem 1. *If the* LWE *assumption holds, the proposed RIBE scheme with En-DKER is selectively secure.*

Proof. We set a series of games, and $\mathcal{A}$'s advantage changes only by a negligible amount between each adjacent games. The first game corresponds to the real selective security for the proposed RIBE scheme, and the final game's ciphertext is independent of the bit b, whereby the advantage of $\mathcal{A}$ is zero. The proof of Theorem 1 is completed.

The Series of Games. Let $\mathcal{A}$ be the adversary in the security definition of the RIBE with En-DKER. We consider the following series of games.

$\mathbf{Game}_0^{(b)}$: This game corresponds to the real selective security game for the proposed RIBE scheme.

$\mathbf{Game}_1^{(b)}$: This game is analogous to $\mathsf{Game}_0^{(b)}$ except the generation of matrices $\boldsymbol{B}$, $\{\boldsymbol{D}_\theta\}_{\theta\in\mathsf{BT}}$, and $\boldsymbol{W}$ during the Setup phase.

1. Select uniformly random matrices $\boldsymbol{R}^*$, $\boldsymbol{S}_\theta^*$ and $\boldsymbol{V}^*$ in $\{-1,1\}^{m\times m}$, where $\theta \in \mathsf{BT}$.[3]
2. Set $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R}^* - \mathsf{H}(\mathsf{ID}^{(b)})\boldsymbol{G}$, $\boldsymbol{W} = \boldsymbol{A}\boldsymbol{V}^* - \mathsf{H}(t^*)\boldsymbol{G}$, and

$$\boldsymbol{D}_\theta = \begin{cases} \boldsymbol{A}\boldsymbol{S}_\theta^*, & \text{if } \theta \in \mathsf{KUNodes}(\mathsf{RL}_{t^*})^*, \\ \boldsymbol{A}\boldsymbol{S}_\theta^* + \boldsymbol{G}, & \text{otherwise.} \end{cases}$$

$\mathbf{Game}_2^{(b)}$: This game is analogous to $\mathsf{Game}_1^{(b)}$ except the generation of the secret key $\mathsf{SK}_{\mathsf{ID}}$ while answering the $\mathcal{Q}_1$ key queries during the Query phase. We divide the generation of $\boldsymbol{sk}'_{\mathsf{ID}}$ and $\boldsymbol{sk}''_{\mathsf{ID},\theta}$ into the following cases, and other steps are the same as $\mathsf{Game}_1^{(b)}$.

- **Case 1**: $\mathsf{ID} = \mathsf{ID}^{(b)}$. In this case, due to the $\mathcal{Q}_1$ key queries restriction in the security definition, the user with the identity ID must have been revoked before the challenge time period t^*. So $\mathsf{Path}(\eta_{\mathsf{ID}}) \cap \mathsf{KUNodes}(\mathsf{RL}_{t^*})^* = \emptyset$, and $\boldsymbol{D}_\theta = \boldsymbol{A}\boldsymbol{S}_\theta^* + \boldsymbol{G}$ for each node $\theta \in \mathsf{Path}(\eta_{\mathsf{ID}})$.
  1. Perform the operation 3.(a) in algorithm **GenSK**.
  2. Sample $\boldsymbol{sk}''_{\mathsf{ID},\theta} \leftarrow \mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{S}_\theta^*, \boldsymbol{G}, \boldsymbol{T}_{\boldsymbol{G}}, \sigma, \bar{\boldsymbol{A}} - \boldsymbol{Z}_{\mathsf{ID}})$, $\theta \in \mathsf{Path}(\eta_{\mathsf{ID}})$.
- **Case 2**: $\mathsf{ID} \neq \mathsf{ID}^{(b)}$ and $\mathsf{Path}(\eta_{\mathsf{ID}}) \cap \mathsf{KUNodes}(\mathsf{RL}_{t^*})^* \neq \emptyset$.
  1. Sample $\boldsymbol{sk}''_{\mathsf{ID},\theta^*} \leftarrow \chi_{\mathsf{LWE}}^{2m\times m}$ and set $\boldsymbol{Z}_{\mathsf{ID}} = [\boldsymbol{A}|\boldsymbol{D}_{\theta^*}]\boldsymbol{sk}''_{\mathsf{ID},\theta^*}$.
  2. $\boldsymbol{sk}''_{\mathsf{ID},\theta} \leftarrow \mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{S}_\theta^*, \boldsymbol{G}, \boldsymbol{T}_{\boldsymbol{G}}, \sigma, \boldsymbol{Z}_{\mathsf{ID}})$, where $\theta \in \mathsf{Path}(\eta_{\mathsf{ID}})(\neq \theta^*)$.
  3. $\boldsymbol{sk}'_{\mathsf{ID}} \leftarrow \mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{R}^*, (\mathsf{H}(\mathsf{ID}) - \mathsf{H}(\mathsf{ID}^{(b)}))\boldsymbol{G}, \boldsymbol{T}_{\boldsymbol{G}}, \sigma, \bar{\boldsymbol{A}} - \boldsymbol{Z}_{\mathsf{ID}})$.
- **Case 3**: $\mathsf{ID} \neq \mathsf{ID}^{(b)}$ and $\mathsf{Path}(\eta_{\mathsf{ID}}) \cap \mathsf{KUNodes}(\mathsf{RL}_{t^*})^* = \emptyset$. In this case, $\boldsymbol{D}_\theta = \boldsymbol{A}\boldsymbol{S}_\theta^* + \boldsymbol{G}$ for each node $\theta \in \mathsf{Path}(\eta_{\mathsf{ID}})$.

[3] This step moves from the algorithm Enc to the Setup phase.

1. Select uniformly random matrix $\boldsymbol{Z}_{\mathsf{ID}}$ in $\mathbb{Z}_q^{n\times m}$ for the identity ID.
2. Sample $\boldsymbol{sk}''_{\mathsf{ID},\theta} \leftarrow \mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{S}^*_\theta, \boldsymbol{G}, \boldsymbol{T}_{\boldsymbol{G}}, \sigma, \boldsymbol{Z}_{\mathsf{ID}})$, where $\theta \in \mathsf{Path}(\eta_{\mathsf{ID}})$.
3. $\boldsymbol{sk}'_{\mathsf{ID}} \leftarrow \mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{R}^*, (\mathsf{H}(\mathsf{ID}) - \mathsf{H}(\mathsf{ID}^{(b)}))\boldsymbol{G}, \boldsymbol{T}_{\boldsymbol{G}}, \sigma, \bar{\boldsymbol{A}} - \boldsymbol{Z}_{\mathsf{ID}})$.

$\mathbf{Game}_3^{(b)}$: This game is analogous to $\mathsf{Game}_2^{(b)}$ except the generation of the decryption key $\mathsf{DK}_{\mathsf{ID},t}$ while answering the $\mathcal{Q}_2$ key queries during the **Query** phase when $\mathsf{ID} = \mathsf{ID}^{(b)}$, $\mathsf{Path}(\eta_{\mathsf{ID}}) \cap \mathsf{KUNodes}(\mathsf{RL}_{t^*})^* \neq \emptyset$ and $t \neq t^*$. [4]

1. Sample $\widetilde{\boldsymbol{sk}}_t \leftarrow \mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{V}^*, (\mathsf{H}(t) - \mathsf{H}(t^*))\boldsymbol{G}, \boldsymbol{T}_{\boldsymbol{G}}, \sigma, \bar{\boldsymbol{A}})$.
2. Perform the operation 2.(a) and 2.(b) in algorithm **GenDK**.
3. Compute $\widetilde{\boldsymbol{k}}''_{i,\mathsf{ID},t} = \widetilde{\boldsymbol{sk}}_t \boldsymbol{k}'_{i,\mathsf{ID},t}$, satisfying $[\boldsymbol{A}|\boldsymbol{W}_t]\widetilde{\boldsymbol{k}}''_{i,\mathsf{ID},t} = \boldsymbol{u}_i - \boldsymbol{h}_{i,\mathsf{ID},t}$, where $\widetilde{\boldsymbol{k}}''_{i,\mathsf{ID},t} \in \mathbb{Z}_q^{2m}$.
4. Split $\boldsymbol{k}_{i,t}$ into four parts, $\boldsymbol{k}_{1,i,t}$, $\boldsymbol{k}_{2,i,t}$, $\boldsymbol{k}_{3,i,t}$, $\boldsymbol{k}_{4,i,t}$, and $\widetilde{\boldsymbol{k}}''_{i,\mathsf{ID},t}$ into two parts $\widetilde{\boldsymbol{k}}''_{1,i,\mathsf{ID},t}$, $\widetilde{\boldsymbol{k}}''_{2,i,\mathsf{ID},t}$, m rows per part. Then, generate

$$\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} = \left[\begin{pmatrix} \boldsymbol{k}_{1,i,t} + \widetilde{\boldsymbol{k}}''_{1,i,\mathsf{ID},t} \\ \boldsymbol{k}_{2,i,t} \end{pmatrix}^\top \middle| \begin{pmatrix} \boldsymbol{k}_{3,i,t} \\ \boldsymbol{k}_{4,i,t} + \widetilde{\boldsymbol{k}}''_{2,i,\mathsf{ID},t} \end{pmatrix}^\top \right]^\top \in \mathbb{Z}_q^{4m}.$$

$\mathbf{Game}_4^{(b)}$: This game is analogous to $\mathsf{Game}_3^{(b)}$ except the generation of the matrix $\boldsymbol{A}$ and the ciphertexts.

1. Select a uniformly random matrix $\boldsymbol{A}$ in $\mathbb{Z}_q^{n\times m}$.
2. Choose $C'_i \leftarrow \mathbb{Z}_q$ and $\boldsymbol{c}_{\mathsf{ID}^{(b)},\theta,t^*} \leftarrow \mathbb{Z}_q^{4m}$, where $\theta \in \mathsf{KUNodes}(\mathsf{RL}_{t^*})^*$, $i \in [l]$.

Analysis Set function $\mathcal{P}_{\mathcal{A},x}(\lambda)$: $\mathbb{N} \to [0,1]$ denote the probability that $\mathcal{A}$ correctly guesses the challenge bit b on input the security parameter $\lambda \in \mathbb{N}$ in the game $\mathsf{Game}_x^{(b)}$. From the definition of $\mathsf{Game}_0^{(b)}$, it follows that the advantage of $\mathcal{A}$ is $\mathsf{Adv}^{\mathsf{SEL\text{-}En\text{-}CPA}}_{\mathsf{RIBE},\mathcal{A}}(\lambda) = |\mathcal{P}_{\mathcal{A},0}(\lambda) - 1/2|$. In addition, $\mathcal{P}_{\mathcal{A},4}(\lambda) = 1/2$ since we make the ciphertext independent of bit b through the LWE assumption in $\mathsf{Game}_4^{(b)}$. So for all $\lambda \in \mathbb{N}$, we have

$$\mathsf{Adv}^{\mathsf{SEL\text{-}En\text{-}CPA}}_{\mathsf{RIBE},\mathcal{A}}(\lambda) \leq \sum_{x\in[4]} |\mathcal{P}_{\mathcal{A},x-1}(\lambda) - \mathcal{P}_{\mathcal{A},x}(\lambda)| \leq \sum_{x\in[4]} \mathsf{negl}_x(\lambda)$$

We will demonstrate that the difference between successive games is only by a negligible amount $\mathsf{negl}_x(\lambda)$, as proven in a series of lemmas in Appendix A.

5 Implementation and Evaluation

In this section, we first compare our scheme with existing revocation models in theory. Then, the performance of our scheme is further evaluated by using simulation experiments.

[4] In this case, challenger $\mathcal{C}$ cannot simulate the secret key $\{\boldsymbol{sk}_{\mathsf{ID},\theta}\}_{\theta\in\mathsf{Path}(\eta_{\mathsf{ID}})}$, but $\mathcal{C}$ can construct a secret key $\widetilde{\boldsymbol{sk}}_t$ that satisfies $[\boldsymbol{A}|\boldsymbol{W}_t]\widetilde{\boldsymbol{sk}}_t = \bar{\boldsymbol{A}}$.

Table 1: Revocation model comparison. Where SK and CT represent the size of secret key and ciphertext, KGC's pw represents the KGC's periodic workload, and RL permission refers to the entity responsible for managing real-time revocation information of users in the system.

Revocation model	SK	KGC's pw	CT	RL permission
Indirect [8]	$O(logN)$	$O(r\log(N/r))$	$O(1)$	KGC
Direct [5]	$O(logN)$	–	$O(r\log(N/r))$	Encryptor
Server-aided [22]	$O(1)$	$O(r\log(N/r))$	$O(1)$	KGC
Ours	$O(logN)$	≈ 0	$O(r\log(N/r))$	KGC

5.1 Theoretical evaluation

As shown in Table 1, we compare our scheme with three existing revocation models, indirect revocation [8], direct revocation [5], and server-aided revocation [22]. It can be observed that our scheme has two main advantages, periodic workload of the KGC is nearly zero, and the encryptor is not required to handle real-time revocation information of users within the system.

5.2 Experimental evaluation

Our scheme runs on a Ubuntu laptop with an AMD Ryzen7 6800HS CPU and 16GB of memory. For better portability, we implement our program using the NTL library and C++ language. We set two sets of parameters: $n = 64$, $m = 390$, $q = 2^{20}$, and $n = 128$, $m = 774$, $q = 2^{23}$.

The Sampling Algorithms. This paper mainly employs three sampling algorithms: TrapGen, SamplePre, and SampleLeft, which are the cornerstone of our scheme and also the most time-consuming in the implementation. To ensure efficient algorithm execution, we concentrate on two optimizations: extracting the Schmidt orthogonalization operation as a preprocessing step to eliminate redundant calculations during each sampling, and harnessing parallel programming to improve computational efficiency. As shown in Table 2, we provide the average runtime of these algorithms over ten executions.

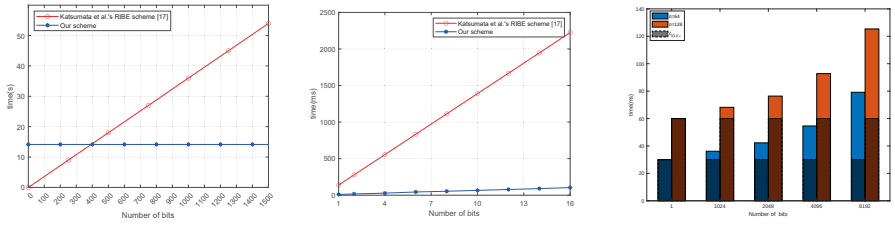
Our scheme. Now, we compare the runtime overhead of our scheme with Katsumata et al.'s lattice-based RIBE scheme with DKER [17]. Our scheme consists of the six algorithms (Setup, GenSK, NodesUp, GenDK, Enc, Dec), where Setup and Dec is similar to other schemes, and we record the runtime in Table 2. The NodesUp algorithm only involves one KUNodes operation, so the runtime is nearly zero.

As shown in Fig 1a, the runtime overhead for the KGC to generate secret keys remains constant in multi-bit encryption. Referring to Fig 1b, as the number of encrypted bits increases, the workload for users to generate decryption keys

Table 2: The running time of sampling, Setup, and Dec algorithms.

Time(ms)	TrapGen	SamplePre	SampleLeft	Setup	Dec
$n = 64$	114	159	167	323	0.1386
$n = 128$	396	314	330	1362	0.342

in our system grows slowly. It only involves some matrix operations, while the time-consuming sampling process is completely outsourced to the server.



(a) GenSK (b) GenDK (c) Enc

Fig. 1: The main runtime of our scheme.

In our scheme, the runtime overhead of the Enc algorithm can be divided into two parts: C_i, which is related to the plaintext, and $\boldsymbol{c}_{\mathsf{ID},\theta,t}$, which is unrelated to the plaintext. As shown in Fig 1c, we set the maximum number of users N is 5000, and the number of revoked users r is 100, the shaded area represents the time overhead of the $\boldsymbol{c}_{\mathsf{ID},\theta,t}$ part of the encryption, which remains constant as the number of encrypted bits increases. Moreover, C_i part takes 0.006ms when encrypting one bit.

6 Conclusion

In this paper, we propose a lattice-based RIBE scheme with En-DKER, which is the first RIBE scheme to ensure confidentiality and anonymity under decryption key exposure. Additionally, we introduce a novel approach to delegate a lattice basis. Leveraging this approach, our scheme significantly reduces the periodic workload for users to generate decryption keys. We prove the security of our scheme under the LWE assumption and efficiently implemented through the NTL and multi-threaded parallel program. The experimental results show that our scheme is suitable for multi-bit encryption and scenarios where the KGC has a high computational workload. Lastly, how to construct an adaptive secure RIBE with En-DKER is the direction of our future research.

A The Series of Lemmas

Lemma 6. *For any adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}_1(\cdot)$ *satisfying* $|\mathcal{P}_{\mathcal{A},0}(\lambda) - \mathcal{P}_{\mathcal{A},1}(\lambda)| \leq \mathsf{negl}_1(\lambda)$.

Proof. The difference between $\mathsf{Game}_0^{(b)}$ and $\mathsf{Game}_1^{(b)}$ is the generation of matrices $\boldsymbol{B}$, $\{\boldsymbol{D}_\theta\}_{\theta\in\mathsf{BT}}$, and $\boldsymbol{W}$. We will analyze these differences individually. For the matrix $\boldsymbol{B}$, by Lemma 4, $\boldsymbol{AR}^*$ is statistically close to the uniform random matrix in $\mathbb{Z}_q^{n\times m}$, and the difference between $\boldsymbol{AR}^*$ and $\boldsymbol{AR}^* - \mathsf{H}(\mathsf{ID}^{(b)})\boldsymbol{G}$ are merely syntactic. It follows that in the adversary's view, the matrix $\boldsymbol{B}$ in $\mathsf{Game}_0^{(b)}$ and $\mathsf{Game}_1^{(b)}$ are statistically indistinguishable. Moreover, the proof of the matrices $\boldsymbol{W}$ and $\{\boldsymbol{D}_\theta\}_{\theta\in\mathsf{BT}}$ are similar. The proof of Lemma 6 is completed.

Lemma 7. *For any adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}_2(\cdot)$ *satisfying* $|\mathcal{P}_{\mathcal{A},1}(\lambda) - \mathcal{P}_{\mathcal{A},2}(\lambda)| \leq \mathsf{negl}_2(\lambda)$.

Proof. The difference between $\mathsf{Game}_1^{(b)}$ and $\mathsf{Game}_2^{(b)}$ is the generation of matrices $\boldsymbol{sk}'_{\mathsf{ID}}$, $\boldsymbol{sk}''_{\mathsf{ID},\theta}$ and $\boldsymbol{Z}_{\mathsf{ID}}$. For the matrix $\boldsymbol{sk}'_{\mathsf{ID}}$, by the properties of sampling algorithms, sampled via algorithm $\mathsf{SampleLeft}$ is statistically close to randomly chosen in $\chi_{\mathsf{LWE}}^{2m\times m}$. Because $\boldsymbol{B}_{\mathsf{ID}} = \boldsymbol{B}+\mathsf{H}(\mathsf{ID})\boldsymbol{G}$, sampled via algorithm $\mathsf{SampleLeft}$ is also statistically close to sampled via algorithm $\mathsf{SampleRight}$. It follows that in the adversary's view, the matrix $\boldsymbol{sk}'_{\mathsf{ID}}$ in $\mathsf{Game}_1^{(b)}$ and the three cases in $\mathsf{Game}_2^{(b)}$ are statistically indistinguishable. The proof of the matrix $\boldsymbol{sk}''_{\mathsf{ID},\theta}$ is similar. So we can also derive that $\boldsymbol{Z}_{\mathsf{ID}} = [\boldsymbol{A}|\boldsymbol{B}_{\mathsf{ID}}]\boldsymbol{sk}'_{\mathsf{ID}}$ and $\boldsymbol{Z}_{\mathsf{ID}} = [\boldsymbol{A}|\boldsymbol{D}_{\theta^*}]\boldsymbol{sk}''_{\mathsf{ID},\theta^*}$ are statistically indistinguishable from a uniformly random matrix selected in $\mathbb{Z}_q^{n\times m}$. The proof of Lemma 7 is completed.

Lemma 8. *For any adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}_3(\cdot)$ *satisfying* $|\mathcal{P}_{\mathcal{A},2}(\lambda) - \mathcal{P}_{\mathcal{A},3}(\lambda)| \leq \mathsf{negl}_3(\lambda)$.

Proof. The difference between $\mathsf{Game}_2^{(b)}$ and $\mathsf{Game}_3^{(b)}$ is the generation of the decryption key $\mathsf{DK}_{\mathsf{ID},t}$. In $\mathsf{Game}_2^{(b)}$ and $\mathsf{Game}_3^{(b)}$,

$$\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} = \left[\begin{pmatrix}\boldsymbol{k}_{1,i,t}+\boldsymbol{k}''_{1,i,\mathsf{ID},\theta^*,t}\\ \boldsymbol{k}_{2,i,t}+\boldsymbol{k}''_{2,i,\mathsf{ID},\theta^*,t}\end{pmatrix}^\top \middle| \begin{pmatrix}\boldsymbol{k}_{3,i,t}+\boldsymbol{k}''_{3,i,\mathsf{ID},\theta^*,t}\\ \boldsymbol{k}_{4,i,t}\end{pmatrix}^\top\right]^\top \in \mathbb{Z}_q^{4m},$$

$$\boldsymbol{dk}_{i,\mathsf{ID},\theta^*,t} = \left[\begin{pmatrix}\boldsymbol{k}_{1,i,t}+\widetilde{\boldsymbol{k}}''_{1,i,\mathsf{ID},t}\\ \boldsymbol{k}_{2,i,t}\end{pmatrix}^\top \middle| \begin{pmatrix}\boldsymbol{k}_{3,i,t}\\ \boldsymbol{k}_{4,i,t}+\widetilde{\boldsymbol{k}}''_{2,i,\mathsf{ID},t}\end{pmatrix}^\top\right]^\top \in \mathbb{Z}_q^{4m},$$

respectively. By the triangle inequality for statistical distance and Lemma 1, since $B > (m\sigma^2+1)2^\lambda$ holds, we can argue that there exists a negligible function $\mathsf{negl}_{smudge}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$\begin{aligned}
&\mathsf{SD}(\boldsymbol{k}_{1,i,t}+\boldsymbol{k}''_{1,i,\mathsf{ID},\theta^*,t}, \boldsymbol{k}_{1,i,t}+\widetilde{\boldsymbol{k}}''_{1,i,\mathsf{ID},t})\\
\leq\ &\mathsf{SD}(\boldsymbol{k}_{1,i,t}+\boldsymbol{k}''_{1,i,\mathsf{ID},\theta^*,t}, \boldsymbol{k}_{1,i,t}) + \mathsf{SD}(\boldsymbol{k}_{1,i,t}, \boldsymbol{k}_{1,i,t}+\widetilde{\boldsymbol{k}}''_{1,i,\mathsf{ID},t})\\
\leq\ &m\cdot\mathsf{negl}_{smudge}(\cdot) + m\cdot\mathsf{negl}_{smudge}(\cdot)\\
=\ &2m\cdot\mathsf{negl}_{smudge}(\cdot).
\end{aligned}$$

The remaining $3m$ dimensional vector proves the same. So in the adversary's view,

$$|\mathcal{P}_{\mathcal{A},2}(\lambda) - \mathcal{P}_{\mathcal{A},3}(\lambda)| \leq 5m \cdot \mathsf{negl}_{smudge}(\cdot).$$

The proof of Lemma 8 is completed.

Lemma 9. *If the* LWE *assumption holds, for any adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}_4(\cdot)$ *satisfying* $|\mathcal{P}_{\mathcal{A},3}(\lambda) - \mathcal{P}_{\mathcal{A},4}(\lambda)| \leq \mathsf{negl}_4(\lambda)$.

Proof. Proof by contradiction, assuming there exists a non-negligible function $\delta(\cdot)$ such that $|\mathcal{P}_{\mathcal{A},3}(\lambda) - \mathcal{P}_{\mathcal{A},4}(\lambda)| \geq \delta(\cdot)$. We can use $\mathcal{A}$ to construct an LWE algorithm $\mathcal{B}$ such that $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(\lambda) \geq \delta(\lambda)$ for all $\lambda \in \mathbb{N}$.

Initial: $\mathcal{A}$ sets the challenge identities $\mathsf{ID}^{(0)}$ and $\mathsf{ID}^{(1)}$, the challenge time period t^*, and the challenge node set $\mathsf{KUNodes}(\mathsf{RL}_{t^*})^*$.

Setup Phase: $\mathcal{B}$ uses $\mathsf{LWE}_{n,q,\sigma}$ challenger to define the matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ and the vector $\boldsymbol{u} \in \mathbb{Z}_q^n$ in public parameters PP. $\mathcal{B}$ makes $m + l$ times queries and receives $\{\boldsymbol{\alpha}_i, \gamma_i\}_{i \in [m+l]} \subset \mathbb{Z}_q^n \times \mathbb{Z}_q$ from $\mathsf{LWE}_{n,q,\sigma}$ challenger, where $\gamma_i = \boldsymbol{s}^\top \boldsymbol{\alpha}_i + e_i \bmod q$. Then set the matrix $\boldsymbol{A} = (\boldsymbol{\alpha}_1 | \cdots | \boldsymbol{\alpha}_m)$ and the vector $\boldsymbol{u}_i = \boldsymbol{\alpha}_{m+i}$, where $i \in [l]$. Other steps are the same as $\mathsf{Game}_3^{(b)}$.

Query Phase: $\mathcal{B}$ replies to the corresponding secret key, decryption key, and revocation queries as in $\mathsf{Game}_3^{(b)}$.

Challenge Phase: $\mathcal{B}$ chooses a random bit $b \leftarrow \{0, 1\}$, and replies based on $\mathsf{ID}^{(b)}$ and t^*. Set $C_i = \gamma_{m+i} + \left\lfloor \frac{q}{2} \right\rfloor \cdot msg_i^{(b)}$ and $\boldsymbol{c}_{\mathsf{ID}^{(b)},\theta,t^*} = \boldsymbol{\gamma}^\top [\boldsymbol{I}_m | \boldsymbol{R}^* | \boldsymbol{S}_\theta^* | \boldsymbol{V}^*]$, where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m) \in \mathbb{Z}_q^m$, $i \in [l]$, and $\theta \in \mathsf{KUNodes}(\mathsf{RL}_{t^*})^*$.

Guess: $\mathcal{A}$ outputs a guess b' of b. Then $\mathcal{B}$ outputs $\mathcal{A}$'s guess as the answer to the $\mathsf{LWE}_{n,q,\sigma}$ challenge.

Note that

$$C_i = \gamma_{m+i} + \left\lfloor \frac{q}{2} \right\rfloor \cdot msg_i^{(b)} = \boldsymbol{s}^\top \boldsymbol{u}_i + \left\lfloor \frac{q}{2} \right\rfloor \cdot msg_i^{(b)} + e_i,$$

$$\boldsymbol{c}_{\mathsf{ID}^{(b)},\theta,t^*} = \boldsymbol{\gamma}^\top [\boldsymbol{I}_m | \boldsymbol{R}^* | \boldsymbol{S}_\theta^* | \boldsymbol{V}^*] = \boldsymbol{s}^\top [\boldsymbol{A} | \boldsymbol{B}_{\mathsf{ID}^{(b)}} | \boldsymbol{D}_\theta | \boldsymbol{W}_{t^*}] + \boldsymbol{e}'^\top [\boldsymbol{I}_m | \boldsymbol{R}^* | \boldsymbol{S}_\theta^* | \boldsymbol{V}^*],$$

where $e_i = e_{m+i}$ and $\boldsymbol{e}' = (e_1, \ldots, e_m)$. So the game simulated by the reduction algorithm $\mathcal{B}$ coincides with $\mathsf{Game}_3^{(b)}$. Simultaneously, based on LWE assumption, C_i and $\boldsymbol{c}_{\mathsf{ID}^{(b)},\theta,t^*}$ are uniformly and independently distributed over $\mathbb{Z}_q$ and $\mathbb{Z}_q^m$, so the game simulated by the reduction algorithm $\mathcal{B}$ coincides with $\mathsf{Game}_4^{(b)}$. Hence, the advantage of $\mathcal{B}$ in solving $\mathsf{LWE}_{n,q,\sigma}$ problem is the same as the advantage of $\mathcal{A}$ in distinguishing $\mathsf{Game}_3^{(b)}$ and $\mathsf{Game}_4^{(b)}$. The proof of Lemma 9 is completed.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (h) ibe in the standard model. In: Eurocrypt. vol. 6110, pp. 553–572. Springer (2010)

2. Ajtai, M.: Generating hard instances of the short basis problem. In: Automata, Languages and Programming: 26th International Colloquium, ICALP'99 Prague, Czech Republic, July 11–15, 1999 Proceedings 26. pp. 1–9. Springer (1999)
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory of Computing Systems **48**, 535–553 (2011)
4. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold fhe. In: Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31. pp. 483–501. Springer (2012)
5. Attrapadung, N., Imai, H.: Attribute-based encryption supporting direct/indirect revocation modes. In: Cryptography and Coding: 12th IMA International Conference, Cryptography and Coding 2009, Cirencester, UK, December 15-17, 2009. Proceedings 12. pp. 278–300. Springer (2009)
6. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7. pp. 566–582. Springer (2001)
7. Benhamouda, F., Gentry, C., Gorbunov, S., Halevi, S., Krawczyk, H., Lin, C., Rabin, T., Reyzin, L.: Can a public blockchain keep a secret? In: Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part I 18. pp. 260–290. Springer (2020)
8. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: Proceedings of the 15th ACM conference on Computer and communications security. pp. 417–426 (2008)
9. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. SIAM journal on computing **32**(3), 586–615 (2003)
10. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Advances in Cryptology-CRYPTO 2006: 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006. Proceedings 26. pp. 290–307. Springer (2006)
11. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. Journal of cryptology **25**, 601–639 (2012)
12. Chen, J., Lim, H.W., Ling, S., Wang, H., Nguyen, K.: Revocable identity-based encryption from lattices. In: Information Security and Privacy: 17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9-11, 2012. Proceedings 17. pp. 390–403. Springer (2012)
13. Emura, K., Takayasu, A., Watanabe, Y.: Adaptively secure revocable hierarchical ibe from k-linear assumption. Designs, Codes and Cryptography **89**(7), 1535–1574 (2021)
14. Ge, A., Wei, P.: Identity-based broadcast encryption with efficient revocation. In: Public-Key Cryptography–PKC 2019: 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part I 22. pp. 405–435. Springer (2019)
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206 (2008)
16. Huang, Z., Lai, J., Han, S., Lyu, L., Weng, J.: Anonymous public key encryption under corruptions. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information

Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part III. pp. 423–453. Springer (2023)
17. Katsumata, S., Matsuda, T., Takayasu, A.: Lattice-based revocable (hierarchical) ibe with decryption key exposure resistance. In: Public-Key Cryptography–PKC 2019: 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II 22. pp. 441–471. Springer (2019)
18. Ling, S., Nguyen, K., Wang, H., Zhang, J.: Revocable predicate encryption from lattices. In: Provable Security: 11th International Conference, ProvSec 2017, Xi'an, China, October 23-25, 2017, Proceedings 11. pp. 305–326. Springer (2017)
19. Luo, F., Al-Kuwari, S., Wang, H., Wang, F., Chen, K.: Revocable attribute-based encryption from standard lattices. Computer Standards & Interfaces **84**, 103698 (2023)
20. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Eurocrypt. vol. 7237, pp. 700–718. Springer (2012)
21. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Advances in Cryptology—CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001 Proceedings 21. pp. 41–62. Springer (2001)
22. Qin, B., Deng, R.H., Li, Y., Liu, S.: Server-aided revocable identity-based encryption. In: Computer Security–ESORICS 2015: 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part I 20. pp. 286–304. Springer (2015)
23. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6), 1–40 (2009)
24. Seo, J.H., Emura, K.: Revocable identity-based encryption revisited: Security model and construction. In: Public-Key Cryptography–PKC 2013: 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26–March 1, 2013. Proceedings 16. pp. 216–234. Springer (2013)
25. Takayasu, A., Watanabe, Y.: Revocable identity-based encryption with bounded decryption key exposure resistance: Lattice-based construction and more. Theoretical Computer Science **849**, 64–98 (2021)
26. Wang, S., Zhang, J., He, J., Wang, H., Li, C.: Simplified revocable hierarchical identity-based encryption from lattices. In: Cryptology and Network Security: 18th International Conference, CANS 2019, Fuzhou, China, October 25–27, 2019, Proceedings 18. pp. 99–119. Springer (2019)
27. Watanabe, Y., Emura, K., Seo, J.H.: New revocable ibe in prime-order groups: adaptively secure, decryption key exposure resistant, and with short public parameters. In: Topics in Cryptology–CT-RSA 2017: The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings. pp. 432–449. Springer (2017)
28. Zhang, Y., Liu, X., Hu, Y.: Simplified server-aided revocable identity-based encryption from lattices. In: Provable and Practical Security: 16th International Conference, ProvSec 2022, Nanjing, China, November 11–12, 2022, Proceedings. pp. 71–87. Springer (2022)