Revocable Identity-based Encryption with Bounded Decryption Key Exposure Resistance: Lattice-based Construction and More*

Atsushi Takayasu † Yohei Watanabe ‡

August 23, 2018

Abstract

In general, identity-based encryption (IBE) does not support an efficient revocation procedure. In ACM CCS'08, Boldyreva et al. proposed *revocable identity-based encryption* (RIBE), which enables us to efficiently revoke (malicious) users in IBE. In PKC 2013, Seo and Emura introduced an additional security notion for RIBE, called *decryption key exposure resistance* (DKER). Roughly speaking, RIBE with DKER guarantees that the security is not compromised even if an adversary gets (a number of) short-term decryption keys. Therefore, DKER captures realistic scenarios and is an important notion.

In this paper, we introduce *bounded decryption key exposure resistance* (B-DKER), where an adversary is allowed to get a-priori bounded number of short-term decryption keys in the security game. B-DKER is a weak version of DKER, but it seems to be sufficient for practical use. We obtain the following results:

- We propose a lattice-based (anonymous) RIBE scheme with B-DKER, which is the first lattice-based construction resilient to decryption key exposure. Our lattice-based construction is secure under the LWE assumption. A previous lattice-based construction satisfies anonymity but is vulnerable even with a single decryption key exposure.
- We propose the first pairing-based RIBE scheme that simultaneously realizes anonymity and B-DKER. Our pairing-based construction is secure under the SXDH assumption.

Our two constructions rely on *cover free families* to satisfy B-DKER, whereas all the existing works rely on *the key re-randomization property* to achieve DKER.

*This paper is the full version of [TW17]. This research was supported by JST CREST Grant Number JPMJCR14D6, Japan, JSPS KAKENHI Grant Number JP17K12697.

†The University of Tokyo, Japan, and National Institute of Advanced Industrial Science and Technology (AIST), Japan. This work was done when the author was JSPS Research Fellow (DC1). e-mail: takayasu@mist.i.u-tokyo.ac.jp

‡The University of Electro-Communications, Japan, and National Institute of Advanced Industrial Science and Technology (AIST), Japan. The author is JSPS Research Fellow (PD).

Contents

1 Introduction **3**
- 1.1 Background . . . 3
- 1.2 Our Contributions . . . 4
- 1.3 Our Approach . . . 6
- 1.4 Roadmap . . . 9

2 Preliminaries **9**
- 2.1 Lattice Preliminaries . . . 9
- 2.2 Pairing Preliminaries . . . 11
- 2.3 KUNode Algorithm . . . 12
- 2.4 Cover Free Families . . . 12

3 B-DKER RIBE **13**
- 3.1 Definitions . . . 13
- 3.2 Treatment of Binary Trees . . . 17
- 3.3 Strategy-Dividing Lemma . . . 17

4 B-DKER RIBE Scheme from Lattices **18**
- 4.1 Construction . . . 18
- 4.2 Security . . . 21
- 4.3 Discussion . . . 27

5 B-DKER RIBE Scheme from Pairings **29**
- 5.1 Construction . . . 29
- 5.2 Security . . . 31
  - 5.2.1 Semi-functional Ciphertexts, Keys, and Key Updates . . . 32
  - 5.2.2 Proof Idea and Game Sequence . . . 33
  - 5.2.3 Proof of Theorem 2 . . . 35

6 Concluding Remarks **43**

A Missing Proofs of Lemmas **49**
- A.1 **Game** 2_{j-1} → **Game** 2_j with $1 \leq j \leq Q_{sk}$ . . . 49
- A.2 **Game** 3_{j-1} → **Game** 3_j with $1 \leq j \leq |\mathcal{T}|$ . . . 54

1 Introduction

1.1 Background

Identity-based encryption (IBE) [Sha84], which is one of the central cryptographic primitives, enables ones to use any strings (e.g., e-mail address) as their public keys. Boneh and Franklin [BF03] proposed the first practical IBE scheme from bilinear groups, and since then, various pairing-based IBE constructions have been proposed (e.g., [Wat05, BW06, Wat09, BB11, RS14, JR17]). Recently, lattice-based schemes [GPV08, ABB10a, Boy10, CHKP12, AFL16, BL16, KY16, Yam16, ZCZ16, Yam17] have attracted attention over the years since lattice-based schemes are believed to resist quantum attacks and the average-case security is guaranteed by the worst-case lattice assumptions.

The most practical point of IBE is that it does not need certificates of public keys. However, since public-key encryption (PKE) provides an efficient revocation procedure by invalidating certificates, the merit of IBE causes another problem: *How do we efficiently revoke (malicious) users in IBE?* Revocation functionality is important to handle users in cryptographic schemes. Boneh and Franklin [BF03] suggested the following naïve revocation procedure: A key generation center (KGC) periodically generates secret keys for all *non-revoked* users and distributes them. However, this solution is inefficient and not scalable since $O(N - r)$ computation is needed every time the KGC updates keys, where N and r are the number of all and revoked users, respectively.

Boldyreva et al. [BGK08] proposed the first (pairing-based) IBE scheme that realizes $O(r \log(N/r))$ computation per update. Such an IBE scheme with a logarithmic revocation procedure is called a *revocable IBE* (RIBE) scheme. Their scheme is based on fuzzy IBE [SW05] and a subset cover framework called the complete subtree (CS) method, which was previously used in the context of broadcast encryption [NNL01]. Specifically, Boldyreva et al. split secret keys into two types: *Long-term secret keys* and *short-term decryption keys*. In every time period, the KGC generates update information called a *key update* (with logarithmic computation), and broadcasts it. If a user is not revoked at the time period, he/she can derive a decryption key for the time period from his/her secret key and the key update received from the KGC. Otherwise (i.e., if the user is revoked), he/she cannot get the decryption key. After Boldyreva et al.'s work, several RIBE schemes have been proposed. Libert and Vergnaud [LV09] proposed the first adaptively secure scheme. Lee et al. [LLP17] proposed the RIBE scheme with the subset difference method. These schemes used pairing. Chen et al. [CLL+12b], Chang et al. [CCKS18], and Hu et al. [HLCL18] proposed the first lattice-based, code-based, and CDH-based (without pairing) RIBE scheme, respectively.

Seo and Emura [SE14b] introduced an additional security notion, called *decryption key exposure resistance* (DKER), and proposed the first RIBE scheme with DKER (DKER RIBE for short) from bilinear groups. DKER literally guarantees that an RIBE scheme is not compromised even if an adversary gets (a number of) decryption keys (i.e., even if polynomially many decryption keys are exposed). Hence, DKER captures a realistic threat, and is an important notion in terms of practical use. Moreover, Boneh-Franklin's naïve solution actually satisfies DKER. Looking back, the motivation of Boldyreva et al. was to make the Boneh-Franklin's revocation procedure efficient, and therefore DKER should be a basic security notion. Indeed, DKER has become the standard security notion of RIBE, and a variety of DKER RIBE schemes [LLP17, ESY16, IWS15, Lee16, LP16, RLPL15, SE16, WES17] have been proposed.

Technically, all the existing DKER RIBE schemes rely on a special property, called *a key re-randomization property*, which is a property that decryption keys can be re-randomized by using only public parameters. Roughly speaking, an adversary cannot derive the long-term secret key from exposed decryption keys (and key updates) since they are re-randomized. The key re-randomization property is quite useful and crucial for the security proof, however it also leads to the following open problems, which were posed by Seo and Emura [SE14b].

1. *Lattice-based DKER RIBE schemes.* As described above, lattice-based cryptography has remarkably developed due to its potentiality for expressive functionality and post-quantum property. However, all previous lattice-based RIBE schemes basically do not have the key re-randomization property due to the way to generating secret keys. In fact, existing RIBE schemes [CLL+12b, CZ15, NWZ16] do not satisfy DKER.[1] In particular, Chen et al.'s RIBE scheme [CLL+12b] immediately becomes insecure even with a single decryption key query (i.e., once any single decryption key is exposed). Hence, the limitation does not stem from proof techniques.

2. *Anonymous DKER RIBE schemes.* As discussed in [BBDP01] in the context of PKE, *anonymity* (or *recipient anonymity*) of IBE/hierarchcial IBE (HIBE) is also an important security notion for practical use. However, as already discussed in the context of IBE/HIBE in [BW06], the anonymity never seems to be achieved if elements for re-randomization are distributed as part of the public parameters. Therefore, it seems difficult to simultaneously realize the key re-randomization property and anonymity. In fact, existing anonymous RIBE schemes [CLL+12a, XWW+17] do not satisfy DKER, and all the existing pairing-based DKER RIBE schemes (e.g., [SE14b, LLP17, WES17]) are non-anonymous.

1.2 Our Contributions

As described above, RIBE should have the resilience property against the leakage of decryption keys, and DKER, i.e., the security of RIBE is nor compromised even if *polynomially many decryption keys* are exposed, is an important and useful security notion. However, there seems to be a gap between DKER and the requirement that we actually need since we can assume that in the real world, the key leakage rarely happens, not often, though the leakage cannot be completely prevented.

Therefore, in this paper, we newly introduce a security notion for RIBE, called *bounded DKER*, to bridge the gap and resolve the above open problems. Bounded DKER is a weak version of DKER. Specifically, bounded DKER guarantees that the security of RIBE is not compromised even if *a-priori bounded number of decryption keys*, which is denoted by Q, are exposed. We believe that bounded DKER is still useful and practical even though the number of exposed decryption keys has to be a-priori bounded. Note that most existing RIBE schemes that do not have DKER are insecure even in the case $Q = 1$, since an adversary can get a long-term secret key with only a single decryption key query.

One may think that the notion of B-DKER RIBE is similar to that of bounded-collusion IBE [GLW12] or k-resilient IBE [HK04]. However, we emphasize that there is a major gap among

[1] Actually, there are unavoidable bugs in the security proof of Cheng and Zhang's RIBE scheme [CZ15] (which we communicated to the authors). See Section 4.3 for the detail.

them and bounded DKER from the practical aspect. In the bounded-collusion IBE, the number of secret key extraction queries is a-priori bounded, whereas our definition allows unbounded collusion, i.e., an adversary can unboundedly issue secret key extraction queries and decryption key queries except for the target identity. Practically, in the bounded-collusion IBE scenario, an adversary might collude with the larger number of users than the a-priori bounded number. The KGC may be unaware of the behind-the-scenes collusion, and thus the system would not be refreshed before breaking it. On the one hand, in the B-DKER RIBE scenario, it would appear that decryption key exposures happen only through human errors or some accident. That is, the leakage cannot be controlled by adversaries. The KGC may notice the fact of leakage from users who are honest but leaked their keys, and therefore the KGC can keep the scheme secure by refreshing it at some point.

Then, we resolve the above open problems of RIBE by not using the key re-randomization technique, but instead weakening the requirement of DKER. More specifically, we propose the following two RIBE schemes with bounded DKER (B-DKER RIBE schemes for short).

Lattice-based Anonymous B-DKER RIBE Scheme. We propose a lattice-based construction with bounded DKER from the learning with errors (LWE) assumption in the selective security model. Our construction is: (1) the first lattice-based RIBE scheme that has the resilience property against decryption key exposure; (2) the first anonymous RIBE scheme with (a kind of) DKER; and (3) can be easily extended to the first lattice-based RIBE scheme in the semi-adaptive security model, where the adversary issues the challenge identity and the challenge time period just after receiving a public parameter.

Pairing-based Anonymous B-DKER RIBE Scheme. We propose a pairing-based construction with bounded DKER from the symmetric external Diffie-Hellman (SXDH) assumption, which is the first pairing-based construction of an anonymous RIBE scheme that has the resilience property against decryption key exposure.

As mentioned above, our two constructions never rely on the key re-randomization, which is the core technique to achieve DKER in previous work. Namely, no useful information is leaked from exposed decryption keys since each decryption key is masked with fresh randomness. Instead, we use *cover free families* (CFFs) to achieve bounded DKER. In a nutshell, CFFs extract a number of subsets from a set, and guarantees that any at most Q subsets do not cover any other subset. where Q is a predetermined parameter. Therefore, we prepare a set of secret keys (for every identity), and extracts a lot of subsets from it. We use different subsets of secret keys to generate each decryption key. In other words, each subset depends on each time period. Consequently, at most Q exposed decryption keys do not reveal any useful information on any other decryption keys due to the underlying CFFs, though the secret key is relatively longer than existing schemes (see Section 1.3 for details).

Improvements from the proceedings version [TW17]. This version, which is an extended version of [TW17], includes the following refinements and new results. We newly propose a pairing-based construction, which did not appear in the conference version in Section 5. Plus, we refine

the syntax and security definition of B-DKER RIBE based on [KMT18] (see Section 3), and our two constructions are proved to be secure in the sense of the rigorous security definition.

1.3 Our Approach

In this section, we show brief overview of our construction; how to achieve B-DKER by using CFFs.

Lattices. At first, we explain our approach to construct lattice-based B-DKER RIBE by modifying Chen et al.'s RIBE scheme [CLL+12b]. The public parameter of Chen et al.'s RIBE scheme consists of three matrices $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2$ and a syndrome vector $\mathbf{u}$ along with a gadget matrix[2] $\mathbf{G}$ that was introduced in [MP12]. The ciphertext of a plaintext $M \in \{0,1\}$ for an identity ID and a time period T is

$$[\mathbf{A}_0|\mathbf{A}_1 + H(\mathsf{ID})\mathbf{G}|\mathbf{A}_2 + H(\mathsf{T})\mathbf{G}]^\top \mathbf{s} + \mathsf{noise} \quad \text{and} \quad \mathbf{u}^\top \mathbf{s} + M \left\lfloor \frac{q}{2} \right\rfloor + \mathsf{noise},$$

where $\mathbf{s}$ is a random secret vector and $H(\cdot)$ is a public hash function. Each user has a long-term secret key $\mathbf{e}'$ whereas KGC broadcasts a key update $\tilde{\mathbf{e}}$ in each time period such that

$$[\mathbf{A}_0|\mathbf{A}_1 + H(\mathsf{ID})\mathbf{G}]\,\mathbf{e}' = \mathbf{u}' \qquad \text{and} \qquad [\mathbf{A}_0|\mathbf{A}_2 + H(\mathsf{T})\mathbf{G}]\,\tilde{\mathbf{e}} = \tilde{\mathbf{u}} \tag{1}$$

for some random syndrome vectors $\mathbf{u}'$ and $\tilde{\mathbf{u}}$. If the user is non-revoked, these two syndrome vectors satisfy $\mathbf{u}' + \tilde{\mathbf{u}} = \mathbf{u}$. The short-term decryption key $\mathbf{e}$ for $(\mathsf{ID}, \mathsf{T})$ is their concatenation $\mathbf{e} := (\mathbf{e}', \tilde{\mathbf{e}})$.

As opposed to an ordinary IBE, the RIBE simulator should create a long-term secret key $\mathbf{e}'$ for the target identity ID^* and a key update $\tilde{\mathbf{e}}$ for the challenge time period T^*. Chen et al. resolved the problem by utilizing a Gaussian sampling algorithm in a clever way. If we do not care about DKER, the simulator should create either a secret key $\mathbf{e}'$ for the target identity ID^* or a key update $\tilde{\mathbf{e}}$ for the target time period T^*. Then, the simulator picks $\mathbf{e}'$ or $\tilde{\mathbf{e}}$ in advance and sets $\mathbf{u}'$ or $\tilde{\mathbf{u}}$ according to the equation (1). Notice that the simulator can create long-term secret keys and key updates for all the other $\mathsf{ID} \neq \mathsf{ID}^*$ and $\mathsf{T} \neq \mathsf{T}^*$ since it has a trapdoor.

In short, to obtain DKER, the challenge ciphertext for the target $(\mathsf{ID}^*, \mathsf{T}^*)$ should not be decrypted by using a key update for T^* and short-term decryption keys for $(\mathsf{ID}^*, \mathsf{T})$ such that $\mathsf{T} \neq \mathsf{T}^*$. However, since Chen et al.'s short-term decryption key is a simple concatenation, the target decryption key for $(\mathsf{ID}^*, \mathsf{T}^*)$ can be recovered even with a single decryption key for $(\mathsf{ID}^*, \mathsf{T})$. Since there is a concrete attack, the limitation is not due to proof techniques but the construction. In other words, the simulator should create both short-term decryption keys $\mathbf{e}'$ for $(\mathsf{ID}^*, \mathsf{T})$ such that $\mathsf{T} \neq \mathsf{T}^*$ and key updates $\tilde{\mathbf{e}}$ for T^* during the simulation. However, once the simulator uses a Gaussian sampling algorithm and sets $\mathbf{e}'$, the corresponding syndrome $\mathbf{u}'$ is fixed. Then, the simulator cannot create $\tilde{\mathbf{e}}$ for $\tilde{\mathbf{u}}$ such that $\mathbf{u}' + \tilde{\mathbf{u}} = \mathbf{u}$.

To resolve the problem, we employ a novel RIBE construction. A starting point of our modification is that our key update $\tilde{\mathbf{e}}$ for a time period T satisfies

$$[\mathbf{A}_0|\mathbf{A}_2 + H(\mathsf{T})\mathbf{G}]\,\tilde{\mathbf{e}} = \tilde{\mathbf{u}}_{\mathsf{T}}$$

[2] Although the gadget matrix was not used by Chen et al. [CLL+12b], it is well known that the parameters can be reduced by utilizing the matrix.

where the corresponding syndrome vector $\tilde{\mathbf{u}}_{\mathsf{T}}$ changes in each time period. The property directly suggests that decryption keys for $(\mathsf{ID}^*, \mathsf{T})$ such that $\mathsf{T} \neq \mathsf{T}^*$ are useless to recover a decryption key for the target $(\mathsf{ID}^*, \mathsf{T}^*)$. However, a new problem occurs by the construction. Since a secret key $\mathbf{e}'$ corresponds to a fixed syndrome vector $\mathbf{u}'$, even non-revoked users cannot derive well-formed decryption keys such that $\mathbf{u}' + \tilde{\mathbf{u}}_{\mathsf{T}} = \mathbf{u}$ for all time periods with their secret keys and key updates. To overcome the issue, in our scheme, each user ID has multiple d secret keys $\mathbf{e}'_1, \ldots, \mathbf{e}'_d$ such that

$$[\mathbf{A}_0 | \mathbf{A}_1 + H(\mathsf{ID})\mathbf{G}]\, \mathbf{e}'_1 = \mathbf{u}'_1, \ldots, [\mathbf{A}_0 | \mathbf{A}_1 + H(\mathsf{ID})\mathbf{G}]\, \mathbf{e}'_d = \mathbf{u}'_d.$$

A naive approach for the scheme to work correctly is that we use each $\mathbf{e}'_\ell$ in each time period. However, the modification makes the scheme too inefficient since the number of secret keys d has to be at least larger than the maximum time period and results in large polynomial. To reduce the size, we set $\mathbf{u} - \tilde{\mathbf{u}}_{\mathsf{T}}$ as a subset sum of $\mathbf{u}'_1, \ldots, \mathbf{u}'_d$ so that non-revoked users can produce well-formed decryption keys with smaller d. The resulting decryption key is a concatenation of the corresponding subset sum of $\mathbf{e}'_1, \ldots, \mathbf{e}'_d$ and the key update $\tilde{\mathbf{e}}$. The simulator utilizes a Gaussian sampling algorithm to create $d-1$ secret key elements $\mathbf{e}'_1, \ldots, \mathbf{e}'_d$ except $\mathbf{e}'_{\ell^*}$ for ID^* and a key update $\tilde{\mathbf{e}}$ for T^* along with their corresponding syndrome vectors, then answers decryption key queries for $(\mathsf{ID}^*, \mathsf{T})$ such that $\mathsf{T} \neq \mathsf{T}^*$. The remaining syndrome vector $\mathbf{u}'_{\ell^*}$ is directly fixed. If $\mathbf{e}'_{\ell^*}$ is not used to answer Q decryption key queries, the approach goes well.

For the above construction to become a provably secure practical RIBE scheme whose adversary is allowed to query Q decryption keys, there are the following three requirements: (1) the number of secret keys d is at most polynomially bounded, (2) a subset sum of $\mathbf{u}_1, \ldots, \mathbf{u}_d$ produces distinct vectors whose number is larger than the maximum time period, (3) there is at least one secret key $\mathbf{e}'_{\ell^*}$ that is not used to answer arbitrary Q decryption key queries. Therefore, we use CFFs so that the resulting scheme satisfies all the above requirements.

Pairing. As mentioned earlier, several pairing-based RIBE schemes that achieve DKER have been already proposed. We here aim to realize an *anonymous* RIBE scheme with (B-)DKER. To obtain small parameters, we utilize dual system encryption methodology. Specifically, our construction is based on a two-level anonymous HIBE scheme proposed by Jutla and Roy [JR17, RS14].

We begin with a basic idea to construct an RIBE scheme. An RIBE decryption key for ID at a time period T is considered as a two-level HIBE secret key for a two-dimension ID vector $(\mathsf{ID}, \mathsf{T})$. Most of existing HIBE schemes realize a two-level secret key for $(\mathsf{ID}_1, \mathsf{ID}_2)$ is in the form of

$$\mathsf{MK} \cdot \mathsf{F}_1(\mathsf{ID}_1)^r \cdot \mathsf{F}_2(\mathsf{ID}_2)^s,$$

where MK is a master key, $\mathsf{F}_i(\cdot)$ for $i = 1, 2$ is a hash function that takes an identity as input, and r and s are randomness. Therefore, an RIBE decryption key for ID at T is constructed in the form of

$$\mathsf{DK}_{\mathsf{ID},\mathsf{T}} := \mathsf{MK} \cdot \mathsf{F}_1(\mathsf{ID})^r \cdot \mathsf{F}_2(\mathsf{T})^s.$$

Hence, we construct an RIBE secret key $\mathsf{SK}_{\mathsf{ID}}$ and key update KU_{T} by dividing the above $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$. Moreover, we add *noise* to them so that $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ can be obtained only from both of appropriate

$\mathsf{SK}_{\mathsf{ID}}$ and KU_{T}. Therefore, $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ cannot be delegated from either $\mathsf{SK}_{\mathsf{ID}}$ or KU_{T} but not both due to the noise. Hence, we construct $\mathsf{SK}_{\mathsf{ID}}$ and KU_{T} in the form of

$$\mathsf{SK}_{\mathsf{ID}} := P \cdot \mathsf{F}_1(\mathsf{ID})^r, \quad \mathsf{KU}_{\mathsf{T}} := P' \cdot \mathsf{MK} \cdot \mathsf{F}_2(\mathsf{T})^s,$$

where P and P' are noises. If a user ID is not revoked at a time period T, then it holds $P' = P^{-1}$, and therefore $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ can be obtained by computing $\mathsf{SK}_{\mathsf{ID}} \cdot \mathsf{KU}_{\mathsf{T}}$. We use the CS method to efficiently manage who is revoked and who is not as in the previous schemes.

However, adding the noise is not sufficient to guarantee DKER since an adversary can get $\mathsf{SK}_{\mathsf{ID}}$ from an exposed decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ and a key update KU_{T} by computing $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}/\mathsf{KU}_{\mathsf{T}}$. All the existing pairing-based schemes (with DKER) rely on *the key re-randomization technique* to avoid this problem. Roughly speaking, an adversary cannot derive $\mathsf{SK}_{\mathsf{ID}}$ since every decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ is re-randomized by a decryption key generation algorithm. However, as mentioned earlier, the technique essentially requires that elements for re-randomization are included in the public parameter, and it forces the resulting scheme to be non-anonymous.[3]

CFFs enable us to get around the obstacle. We use the CFF instead of the key re-randomization technique, and finally achieve both B-DKER and anonymity. We first explain how we construct our scheme using the CFF. We prepare a lot of secret keys for ID with different noises,

$$\mathsf{SK}_{\mathsf{ID}} := \{\mathtt{sk}_{\mathsf{ID}}^{(i)}\}_{i=1}^{d} = \{P_i \cdot \mathsf{F}_1(\mathsf{ID})^{r_i}\}_{i=1}^{d},$$

where d depends on a parameter of the underlying CFF, $P_1, \ldots, P_d$ are noises, and $r_1, \ldots, r_d$ are randomness. On the other hand, a key update at T is constructed as follows:

$$\mathsf{KU}_{\mathsf{T}} := \left(\prod_{i \in \mathcal{F}_{\mathsf{T}}} P_i'^{-1} \right) \cdot \mathsf{MK} \cdot \mathsf{F}_2(\mathsf{T})^s,$$

where $\mathcal{F}_{\mathsf{T}}$ is a subset of $\{1, \ldots, d\}$, which is determined by the underlying CFF, and $\{P_i'\}_{i \in \mathcal{F}_{\mathsf{T}}}$ are noises. If a user ID is not revoked at T, then $\mathsf{SK}_{\mathsf{ID}}$ shares the same noise with KU_{T} (i.e., $P_i = P_i'$ for every $i \in \{1, \ldots, d\}$), and hence the user obtains a decryption key for ID at T by computing

$$\mathsf{DK}_{\mathsf{ID},\mathsf{T}} = \prod_{i \in \mathcal{F}_{\mathsf{T}}} \mathtt{sk}_{\mathsf{ID}}^{(i)} \cdot \mathsf{KU}_{\mathsf{T}} = \mathsf{MK} \cdot \mathsf{F}_1(\mathsf{ID})^{\sum_{i \in \mathcal{F}_{\mathsf{T}}} r_i} \cdot \mathsf{F}_2(\mathsf{T})^s.$$

In the security proof, we employ the dual system encryption methodology [Wat09]. Namely, we define *semi-functional forms* of ciphertexts, secret keys, key updates, and decryption keys, and change them from normal forms to semi-functional ones one by one. However, in general, the dual system encryption methodology works well since an adversary cannot obtain a secret key $\mathsf{SK}_{\mathsf{ID}^*}$ for the target identity ID^*, whereas in the RIBE setting, the adversary can get it as well as a key update $\mathsf{KU}_{\mathsf{T}^*}$ at the target time period T^*. In other words, $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$ may leak some important information during the ordinary dual system encryption proof. Therefore, we have to take into account the difference between (H)IBE and RIBE when we design hybrid games in the context of RIBE.

[3] The fact is well known in the context of anonymous HIBE (e.g., [BW06])

To resolve this issue, we take the following proof strategy: We transform $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$ into their semi-functional forms by using the underlying CFF and semi-functional randomness of all other secret keys and key updates. Namely, we first change all secret keys for ID ($\neq \mathsf{ID}^*$) and key updates for T ($\neq \mathsf{T}^*$) (as well as challenge ciphertexts) with semi-functional ones. The underlying CFF ensures that there exists $\mathtt{sk}_{\mathsf{ID}^*}^{(\ell)}$ not used for at most Q decryption keys for ID^*, which are issued to the challenger by the adversary. The semi-functional randomness and the existence of $\mathtt{sk}_{\mathsf{ID}^*}^{(\ell)}$ allow us to transform $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$ into their semi-functional forms without leaking any important information (see Lemma 9 for details). The rest of the proof can be proved by the ordinary dual system encryption methodology. We more rigorously prove the security through the methodology than previous works [CLL+12a, SE15, Lee16, XWW+16, XWW+17, XWWT18]. For instance, we explicitly check that the distributions of semi-functional randomness are correctly simulated, while the previous works do not care about them despite its importance in the security proof.

1.4 Roadmap

In Section 3, we define B-DKER RIBE. In Section 4, we construct the first B-DKER RIBE scheme from the LWE assumption. In Section 5, we construct an anonymous B-DKER RIBE scheme from pairings.

2 Preliminaries

"Probabilistic polynomial-time" is abbreviated as "PPT". We denote $[a, b]$ by a set $\{a, a+1, \ldots, b\}$ for any integers $a, b \in \mathbb{N}$ such that $a \leq b$. We sometimes write $[d]$ as $[1, d]$ for simplicity. Let a bold capital $\mathbf{A}$ and a bold lower $\mathbf{a}$ denote a matrix and a column vector respectively. Let $\mathbf{A}^\top$ and $\mathbf{a}^\top$ denote their transposes. If we write $(y_1, y_2, \ldots, y_m) \leftarrow \mathcal{A}(x_1, x_2, \ldots, x_n)$ for an algorithm $\mathcal{A}$ having n inputs and m outputs, it means to input $x_1, x_2, \ldots, x_n$ into $\mathcal{A}$ and to get the resulting output $y_1, y_2, \ldots, y_m$. If $\mathcal{X}$ is a set, we write $x \xleftarrow{\$} \mathcal{X}$ to mean the operation of picking an element x of $\mathcal{X}$ uniformly at random. We use λ as a security parameter. For sufficiently large λ, a function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is negligible if $\mathsf{negl}(\lambda) < 1/p(\lambda)$ for any polynomial $p(\lambda)$. Let X and Y be two random variables taking values in some finite set Ω. Statistical distance is defined as $\Delta(X; Y)$, as $\Delta(X; Y) := \frac{1}{2} \sum_{s \in \Omega} |\Pr[X = s] - \Pr[Y = s]|$. For sets of random variables X and Y, we say that X and Y are statistically close if $\Delta(X; Y)$ is negligible.

2.1 Lattice Preliminaries

An m-dimensional integer lattice is an additive discrete subgroup of $\mathbb{Z}^m$. For positive integers q, n, m, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and a vector $\mathbf{u} \in \mathbb{Z}_q^m$, the m-dimensional integer (shifted) lattices $\Lambda_q^\perp(\mathbf{A})$ and $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ are defined as $\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{0} \bmod q\}$ and $\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{u} \bmod q\}$. The lattice $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a shift of the lattice $\Lambda_q^\perp(\mathbf{A})$; if $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ then $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^\perp(\mathbf{A}) + \mathbf{t}$. Let $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ be a basis of a lattice $\Lambda_q^\perp(\mathbf{A})$. Then $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ is also a basis of a lattice $\Lambda_q^\perp(\mathbf{H}\mathbf{A})$ for a full rank $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$.

Matrix Norms. For a vector $\mathbf{u}$, we let $\|\mathbf{u}\|$ denote its L_2 norm. For a matrix $\mathbf{R} \in \mathbb{Z}^{k \times m}$, we let $\|\mathbf{R}\|$ denotes the L_2 length of the longest column of $\mathbf{R}$ and $\|\mathbf{R}\|_{\mathsf{GS}} = \|\tilde{\mathbf{R}}\|$ where $\tilde{\mathbf{R}}$ is the Gram-Schmidt orthogonalization of $\mathbf{R}$.

Gaussian Distributions. Let $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ denote the discrete gaussian distribution over Λ with center $\mathbf{c}$ and a parameter σ. If $\mathbf{c} = 0$, we omit the subscript and denote $\mathcal{D}_{\Lambda,\sigma}$. We summarize some basic properties of discrete Gaussian distributions.

Lemma 1 ([GPV08])**.** *Let Λ be an m-dimensional lattice. Let $\mathbf{T}$ be a basis for Λ, and suppose $\sigma \geq \|\mathbf{T}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log m})$. Then $\Pr[\|\mathbf{x}\| > \sigma\sqrt{m} : \ \mathbf{x} \leftarrow \mathcal{D}_{\Lambda,\sigma}] \leq \mathsf{negl}(m)$.*

Lemma 2 ([GPV08])**.** *Let n and q be positive integers with q prime, and let $m \geq 2n \log q$. Then for all but a $2q^{-n}$ fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and for any $\sigma \geq \omega(\sqrt{\log m})$, the distribution of $\mathbf{u} = \mathbf{Ae} \mod q$ is statistically close to uniform over $\mathbb{Z}_q^n$ where $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$. Furthermore, the conditional distribution of $\mathbf{e}$ given $\mathbf{Ae} = \mathbf{u} \mod q$ is exactly $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$.*

Sampling Algorithms.

Lemma 3. *Let $n, m, q > 0$ be positive integers with q prime. There are probabilistic polynomial time algorithms such that*

- *([BLP+13]):* $\mathsf{SampleGaussian}(\mathbf{T}, \sigma) \to \mathbf{e}$
  a randomized algorithm that, given a basis $\mathbf{T}$ for an m-dimensional lattice Λ and a parameter $\sigma \geq \|\mathbf{T}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log m})$ as inputs, then outputs $\mathbf{e}$ which is distributed according to $\mathcal{D}_{\Lambda,\sigma}$.
- *([Ajt99, AP11, MP12]):* $\mathsf{TrapGen}(q, n, m) \to (\mathbf{A}, \mathbf{T_A})$
  a randomized algorithm that, when $m \geq 2n\lceil \log q \rceil$, outputs a full rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ for $\Lambda_q^{\perp}(\mathbf{A})$ such that $\mathbf{A}$ is statistically close to uniform and $\|\mathbf{T_A}\|_{\mathsf{GS}} = O(\sqrt{n \log q})$ with overwhelming probability in n.
- *([CHKP12]):* $\mathsf{SampleLeft}(\mathbf{A}, \mathbf{F}, \mathbf{u}, \mathbf{T_A}, \sigma) \to \mathbf{e}$
  a randomized algorithm that, given a full rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a matrix $\mathbf{F} \in \mathbb{Z}_q^{n \times m}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, a basis $\mathbf{T_A}$ for $\Lambda_q^{\perp}(\mathbf{A})$, and a Gaussian parameter $\sigma > \|\mathbf{T_A}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log m})$ as inputs, then outputs a vector $\mathbf{e} \in \mathbb{Z}_q^{2m}$ sampled from a distribution that is statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}|\mathbf{F}),\sigma}$.
- *([ABB10a]):* $\mathsf{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}, \mathbf{u}, \mathbf{T_G}, \sigma) \to \mathbf{e}$ *where $\mathbf{F} = \mathbf{AR} + \mathbf{G}$*
  a randomized algorithm that, given full rank matrices $\mathbf{A}, \mathbf{G} \in \mathbb{Z}_q^{n \times m}$, a matrix $\mathbf{R} \in \mathbb{Z}^{m \times m}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, a basis $\mathbf{T_G}$ of $\Lambda_q^{\perp}(\mathbf{G})$, and a Gaussian parameter $\sigma > \|\mathbf{T_G}\|_{\mathsf{GS}} \cdot \|\mathbf{R}\| \cdot \omega(\sqrt{\log m})$ as inputs, then outputs a vector $\mathbf{e} \in \mathbb{Z}_q^{2m}$ sampled from a distribution that is statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}|\mathbf{F}),\sigma}$.
- *([MP12]): Let $m > n\lceil \log q \rceil$. Then there is a fixed full rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ such that the lattice $\Lambda_q^{\perp}(\mathbf{G})$ has a publicly known basis $\mathbf{T_G} \in \mathbb{Z}_q^{m \times m}$ with $\|\mathbf{T_G}\|_{\mathsf{GS}} \leq \sqrt{5}$.*

We sometimes call $\mathbf{G}$ a gadget matrix that enables us to reduce several parameters. We use $\mathsf{SampleGaussian}(\mathbf{T}, \sigma)$ only for sampling a distribution $\mathcal{D}_{\mathbb{Z}^m,\sigma}$. For the purpose, we always use a standard basis for $\mathbb{Z}^m$ as $\mathbf{T}$. Hence, we write $\mathsf{SampleGaussian}(\mathbb{Z}^m, \sigma)$ throughout the paper.

To obtain a lower bound of σ, we will use the following fact.

Lemma 4 ([ABB10a]). *Let $\mathbf{R}$ be a $m \times m$ matrix chosen at random from $\{-1, 1\}^{m\times m}$. Then there is a universal constant C such that $\Pr[\|\mathbf{R}\| > C\sqrt{m}] < e^{-m}$.*

Randomness Extraction.

Lemma 5 ([ABB10a]). *Suppose that $m > (n+1)\log_2 q + \omega(\log n)$ and that $q > 2$ is prime. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly in $\{-1, 1\}^{m\times k}$ where $k = k(n)$ is polynomial in n. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n\times m}$ and $\mathbb{Z}_q^{n\times k}$ respectively. Then, for all vectors $\mathbf{e} \in \mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top\mathbf{e})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top\mathbf{e})$.*

Encoding Identities as Matrices.

Definition 1. *Let q be a prime and n be a positive integer. We say that a function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is a full-rank difference (FRD) map if:*

1. *for all distinct $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, the matrix $H(\mathbf{u}) - H(\mathbf{v}) \in \mathbb{Z}_q^{n\times n}$ is full rank,*
2. *H is computable in polynomial time in $n \log q$.*

Learning with Errors (LWE). The security of our RIBE scheme is reduced to the following LWE assumption.

Assumption 1 (Learning with Errors (LWE) Assumption [Reg05]). *For integers $n, m = m(n)$, $\alpha \in (0, 1)$ such that a prime $q = q(n) > 2$ and $\alpha q > 2\sqrt{n}$, define the distribution: $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{x} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\alpha q}, \mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$. We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0, 1\}$), $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{LWE}} := \left|\Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{A}^\top\mathbf{s} + \mathbf{x}) = 1\right] - \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{v}) = 1\right]\right|$ is negligible in the security parameter n.*

Regev [Reg05] showed that (through a quantum reduction) the LWE problem is as hard as approximating the worst-case GapSVP to $\tilde{O}(n/\alpha)$ factors. Peikert [Pei09], Brakerski et al. [BLP+13] showed analogous results through classical reductions.

2.2 Pairing Preliminaries

Bilinear Groups. A bilinear group generator $\mathcal{G}$ is an algorithm that takes a security parameter λ as input and outputs a bilinear group $(q, G_1, G_2, G_T, g_1, g_2, e)$, where q is a prime, G_1, G_2, and G_T are multiplicative cyclic groups of order q, g_1 and g_2 are (random) generators of G_1 and G_2, respectively, and e is an efficiently computable and non-degenerate bilinear map $e : G_1 \times G_2 \to G_T$ with the following bilinear property: For any $u, v \in \mathbb{Z}_q$, $e(g_1^u, g_2^v) = e(g_1, g_2)^{uv}$. In this paper, we consider asymmetric pairing , i.e., $G_1 \neq G_2$.

Complexity Assumptions. We describe the symmetric external Diffie-Hellman (SXDH) assumption, which is the underlying assumption of our pairing-based construction.

Assumption 2 (Decisional DIffie-Hellman Assumption in G_i (DDHi Assumption))**.** *Given a group generator $\mathcal{G}$ with input a security parameter λ, define the distribution:*

$$\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow \mathcal{G}(\lambda), \quad c_1, c_2, \mu \xleftarrow{\$} \mathbb{Z}_q,$$
$$D := (\mathbb{G}, g_i^{c_1}, g_i^{c_2}), \quad V := g_i^{c_1 c_2}, \quad W := g_i^{c_1 c_2 + \mu}.$$

We assume that for any PPT algorithm $\mathcal{A}$ (with output in $\{0,1\}$),

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{DDH}i} := |\Pr[\mathcal{A}(D, V) = 1] - \Pr[\mathcal{A}(D, W) = 1]|$$

is negligible in the security parameter λ.

Assumption 3 (Symmetric External DIffie-Hellman (SXDH) Assumption)**.** *Given a group generator $\mathcal{G}$, both the DDH1 and DDH2 assumptions hold.*

2.3 KUNode Algorithm

To reduce costs of a revocation process, we use a binary tree structure and apply the following KUNode algorithm as in the previous RIBE schemes [BGK08, LV09, SE14b]. $\mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}_\mathsf{T})$ takes as input a binary tree BT and a revocation list RL, and outputs a set of nodes. When η is a non-leaf node, then we write η_L and η_R as the left and right child of η, respectively. When η is a leaf node, $\mathsf{Path}(\mathsf{BT}, \eta)$ denotes the set of nodes on the path from η to the root. Each user is assigned to a leaf node. If a user who is assigned to η is revoked on a time period $\mathsf{T} \in \mathcal{T}$, then $(\eta, \mathsf{T}) \in \mathsf{RL}_\mathsf{T}$. $\mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}_\mathsf{T})$ is executed as follows. It sets $\mathcal{X} := \emptyset$ and $\mathcal{Y} := \emptyset$. For any $(\eta_i, \mathsf{T}_i) \in \mathsf{RL}_\mathsf{T}$, if $\mathsf{T}_i \leq \mathsf{T}$ then it adds $\mathsf{Path}(\mathsf{BT}, \eta_i)$ to $\mathcal{X}$ (i.e., $\mathcal{X} := \mathcal{X} \cup \mathsf{Path}(\mathsf{BT}, \eta_i)$). Then, for any $\eta \in \mathcal{X}$, if $\eta_L \notin \mathcal{X}$, then it adds η_L to $\mathcal{Y}$. If $\eta_R \notin \mathcal{X}$, then it adds η_R to $\mathcal{Y}$. Finally, it outputs $\mathcal{Y}$ if $\mathcal{Y} \neq \emptyset$. If $\mathcal{Y} = \emptyset$, then it adds root to $\mathcal{Y}$ and outputs $\mathcal{Y}$.

2.4 Cover Free Families

We define a cover free family (CFF), which is a core building block in our construction, as follows.

Definition 2 (Cover Free Families [EFF85])**.** *Let α, d, Q be positive integers, and $\mathcal{F} := \{\mathcal{F}_\mu\}_{\mu \in [\alpha]}$ be a family of subsets of $[d]$, where every $|\mathcal{F}_\mu| = w$. $\mathcal{F}$ is said to be w-uniform Q-cover-free if it holds that $\bigcup_{j=1}^{Q} \mathcal{F}_{i_j} \not\supset \mathcal{F}_{i_{Q+1}}$ for any $\mathcal{F}_{i_1}, \mathcal{F}_{i_2}, \ldots, \mathcal{F}_{i_{Q+1}} \in \mathcal{F}$ such that $\mathcal{F}_{i_k} \neq \mathcal{F}_{i_\ell}$ for any distinct $k, \ell \in [Q+1]$.*

Lemma 6 ([KRS99])**.** *There is a deterministic polynomial time algorithm $\mathsf{CFF.Gen}$ that, on input of positive integers α and Q, returns $d \in \mathbb{N}$ and a family $\mathcal{F} = \{\mathcal{F}_\mu\}_{\mu \in [\alpha]}$, such that $\mathcal{F}$ is Q-cover free over $[d]$ and w-uniform, where $d \leq 16Q^2 \log \alpha$ and $w = d/4Q$.*

In our B-DKER RIBE constructions, we assume $\mathcal{F}_\mathsf{T}$ is a d-bit string such that ℓ-th bit is one for $\ell \in \mathcal{F}_\mathsf{T}$ and other bits are zero.

3 B-DKER RIBE

3.1 Definitions

In this section, we formally define *B-DKER RIBE*. $\mathcal{M}$, $\mathcal{I}$, and $\mathcal{T}$ denote sets of plaintexts, identities, and time-periods, respectively. Our definition follows that of recently proposed Katsumata et al.'s R(H)IBE [KMT18]. Although syntaxes and security models of R(H)IBE are slightly different in other papers, we believe that Katsumata et al. introduced the simplest syntax and the most rigorous security model.

Syntax. In the syntax, unlike those in other papers, the state information and the revocation list are parts of the master secret key. Hence, the master secret key may be updated during the secret key generations and key update information generations. Furthermore, we omit the "revocation" algorithm in the syntax since the operation is simple and common in all R(H)IBE schemes; given an identity ID which will be revoked and the time period T, KGC adds a tuple $(\mathsf{ID}, \mathsf{T})$ to the revocation list. Therefore, the revocation list RL_{T} at the time period T contains tuples $(\mathsf{ID}_i, \mathsf{T}_i)$'s, where the ID_i have already been revoked at the time period T_i.

An RIBE scheme Π consists of the six algorithms $(\mathsf{SetUp}, \mathsf{PKG}, \mathsf{KeyUp}, \mathsf{DKG}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:

$\mathsf{SetUp}(1^\lambda) \to (\mathsf{PP}, \mathsf{MK})$: This is the probabilistic *setup* algorithm that takes a security parameter 1^λ as input, and outputs a public parameter PP and a master secret key MK that contains a state st and an initial revocation list $\mathsf{RL}_1 := \emptyset$.

We assume that the plaintext space $\mathcal{M}$, the time period space $\mathcal{T}$, and the identity space $\mathcal{I}$ are determined only by the security parameter λ, and their descriptions are contained in PP.

$\mathsf{PKG}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID}) \to (\mathsf{SK}_{\mathsf{ID}}, \mathsf{MK}')$: This is the *secret key generation* algorithm that takes the public parameter PP, the master secret key MK, and an identity $\mathsf{ID} \in \mathcal{I}$ as input, and may update the state information st in the master secret key MK. Then, it outputs a secret key $\mathsf{SK}_{\mathsf{ID}}$ for the identity ID and also the "updated" master secret key MK'.

$\mathsf{KeyUp}(\mathsf{PP}, \mathsf{T}, \mathsf{MK}) \to (\mathsf{KU}_{\mathsf{T}}, \mathsf{MK}')$: This is the *key update information generation* algorithm that takes the public parameter PP, a time period $\mathsf{T} \in \mathcal{T}$, the master secret key MK (that contains the revocation list RL_{T}) as input, and may update the state information st in MK. Then, it outputs a key update KU_{T} and also the "updated" master secret key MK'.

$\mathsf{DKG}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{T}}) \to \mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ or $\perp$: This is the *decryption key generation* algorithm that takes the public parameter PP, a secret key $\mathsf{SK}_{\mathsf{ID}}$, and a key update KU_{T} as input, and outputs a decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ for the time period T or the special "invalid" symbol $\perp$ if ID has been revoked by T.

$\mathsf{Enc}(\mathsf{PP}, \mathsf{ID}, \mathsf{T}, M) \to \mathsf{CT}_{\mathsf{ID},\mathsf{T}}$: This is the *encryption* algorithm that takes the public parameter PP, an identity ID, a time period T, and a plaintext M as input, and outputs a ciphertext $\mathsf{CT}_{\mathsf{ID},\mathsf{T}}$.

$\mathsf{Dec}(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{T}}, \mathsf{CT}) \to M$: This is the *decryption* algorithm that takes the public parameter PP, a decryption key $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ and a ciphertext CT as input, and outputs the decryption result M.

Correctness. We require the following to hold for an RIBE scheme. Informally, we require a ciphertext corresponding to an identity ID for a time period T to be properly decrypted by ID if ID is not revoked on time T. To fully capture this, we consider all the possible scenarios of creating the secret key for ID. Namely, for all $\lambda \in \mathbb{N}$, $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{SetUp}(1^\lambda)$, $\mathsf{ID} \in \mathcal{I}$, $\mathsf{T} \in \mathcal{T}$, $M \in \mathcal{M}$, and $\mathsf{RL}_\mathsf{T} \subseteq \mathcal{I} \setminus \{\mathsf{ID}\}$, we require $M' = M$ to hold after executing the following procedures:

(1) $(\mathsf{SK}_\mathsf{ID}, \mathsf{MK}) \leftarrow \mathsf{PKG}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$.

(2) $(\mathsf{KU}_\mathsf{T}, \mathsf{MK}) \leftarrow \mathsf{KeyUp}(\mathsf{PP}, \mathsf{T}, \mathsf{MK})$.

(3) $\mathsf{DK}_{\mathsf{ID},\mathsf{T}} \leftarrow \mathsf{DKG}(\mathsf{PP}, \mathsf{SK}_\mathsf{ID}, \mathsf{KU}_\mathsf{T})$.

(4) $\mathsf{CT}_{\mathsf{ID},\mathsf{T}} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ID}, \mathsf{T}, M)$.

(5) $M' \leftarrow \mathsf{Dec}(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{T}}, \mathsf{CT}_{\mathsf{ID},\mathsf{T}})$.

We note that, the most stringent way to define correctness would be to also capture the fact that the secret key MK could be updated after executing PKG. In particular, the output of KeyUp, which takes as input the KGC's secret key MK, may differ in general before and after PKG is run. Therefore, to be more precise, we should also allow an arbitrary (polynomial) number of executions of PKG in between steps (1) and (2). However, we defined correctness as above for the sake of simplicity and readability. We note that our scheme satisfies the more stringent correctness (which will be obvious from construction).

Security Definitions. Here, we give the security definitions of an RIBE scheme $\Pi = (\mathsf{SetUp}, \mathsf{PKG}, \mathsf{KeyUp}, \mathsf{DKG}, \mathsf{Enc}, \mathsf{Dec})$ with *bounded decryption key exposure resistance* (B-DKER). We first give the formal definition of selective-identity security which we call IND-sRID-Q-CPA security via a game between an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. The game is parametrized by the security parameter λ, and has the global counter $\mathsf{T}_{\mathtt{cu}}$, initialized with 1, that denotes the "current time period" with which $\mathcal{C}$'s responses to $\mathcal{A}$'s queries are controlled. Intuitively, $\mathcal{A}$ can get all the key updates, secret keys, and decryption keys under the following restriction about the challenge identity ID^*: Only if ID^* is revoked by the challenge time period T^*, then $\mathcal{A}$ is allowed to get a secret key for ID^*. Otherwise, $\mathcal{A}$ is allowed to get at most Q decryption keys for ID^*. Formally, the game proceeds as follows:

At the beginning, $\mathcal{A}$ sends the challenge identity/time period pair $(\mathsf{ID}^*, \mathsf{T}^*) \in \mathcal{I} \times \mathcal{T}$ to $\mathcal{C}$. Next, $\mathcal{C}$ runs $(\mathsf{PP}, \mathsf{MK}) \leftarrow \mathsf{SetUp}(1^\lambda)$, and prepares a list $\mathtt{SKList}$ that initially contains $(\mathsf{KGC}, \mathsf{MK})$, and into which identity/secret key pairs $(\mathsf{ID}, \mathsf{SK}_\mathsf{ID})$ generated during the game will be stored. From this point on, whenever a new secret key SK_ID is generated for an identity $\mathsf{ID} \in \mathcal{I}$ due to the execution of PKG or the master secret key MK is updated due to the execution of KeyUp, $\mathcal{C}$ will store $(\mathsf{ID}, \mathsf{SK}_\mathsf{ID})$ or update $(\mathsf{KGC}, \mathsf{MK})$ in $\mathtt{SKList}$, and we will not explicitly mention this addition/update. Then,

$\mathcal{C}$ executes $(\mathsf{KU}_1, \mathsf{MK}') \leftarrow \mathsf{KeyUp}(\mathsf{PP}, \mathsf{T}_{\mathsf{cu}} = 1, \mathsf{MK})$, where $\mathsf{RL}_1 = \emptyset$, for generating the initial time period $\mathsf{T}_{\mathsf{cu}} = 1$. After that, $\mathcal{C}$ gives PP and KU_1 to $\mathcal{A}$.

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

Secret Key Generation Query: Upon a query $\mathsf{ID} \in \mathcal{I}$ from $\mathcal{A}$, where it is required that $(\mathsf{ID}, *) \notin \mathtt{SKList}$, $\mathcal{C}$ executes $(\mathsf{SK}_{\mathsf{ID}}, \mathsf{MK}') \leftarrow \mathsf{PKG}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$ (and returns nothing to $\mathcal{A}$).

Secret Key Reveal Query: Upon a query $\mathsf{ID} \in \mathcal{I}$ from $\mathcal{A}$, $\mathcal{C}$ checks if the following conditions are simultaneously satisfied:

(a) $(\mathsf{ID}, \mathsf{SK}_{\mathsf{ID}}) \in \mathtt{SKList}$.

(b) If $\mathsf{T}_{\mathsf{cu}} \geq \mathsf{T}^*$ and $\mathsf{ID}^* \notin \mathsf{RL}_{\mathsf{T}^*}$, then $\mathsf{ID} \neq \mathsf{ID}^*$.

If these conditions are *not* satisfied, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ finds $\mathsf{SK}_{\mathsf{ID}}$ from $\mathtt{SKList}$, and returns it to $\mathcal{A}$.

Revocation & Key Update Query: Upon a query $\mathsf{RL} \subseteq \mathcal{I}$ (which denotes the set of identities that are going to be revoked in the next time period) from $\mathcal{A}$, $\mathcal{C}$ checks if the following conditions are satisfied simultaneously:[4]

(a) $\mathsf{RL}_{\mathsf{T}_{\mathsf{cu}}} \subseteq \mathsf{RL}$.

(b) If $\mathsf{T}_{\mathsf{cu}} = \mathsf{T}^* - 1$ and $\mathsf{SK}_{\mathsf{ID}^*}$ for the challenge ID^* has been revealed to $\mathcal{A}$ via a secret key reveal query ID^*, then $\mathsf{ID}^* \in \mathsf{RL}$.

(c) If $\mathsf{T}_{\mathsf{cu}} = \mathsf{T}^* - 1$ and $\mathsf{DK}_{\mathsf{ID}^*,\mathsf{T}}$ for the challenge ID^* has been revealed to $\mathcal{A}$ more than Q times via decryption key reveal queries $(\mathsf{ID}^*, \mathsf{T})$, then $\mathsf{ID}^* \in \mathsf{RL}$.

If these conditions are *not* satisfied, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$.

Otherwise $\mathcal{C}$ increments the current time period by $\mathsf{T}_{\mathsf{cu}} \leftarrow \mathsf{T}_{\mathsf{cu}} + 1$. Then, $\mathcal{C}$ updates the master secret key MK by setting $\mathsf{RL}_{\mathsf{T}_{\mathsf{cu}}} \leftarrow \mathsf{RL}$, and runs $(\mathsf{KU}_{\mathsf{T}_{\mathsf{cu}}}, \mathsf{MK}') \leftarrow \mathsf{KeyUp}(\mathsf{PP}, \mathsf{T}_{\mathsf{cu}}, \mathsf{MK})$. Finally, $\mathcal{C}$ returns $\mathsf{KU}_{\mathsf{T}_{\mathsf{cu}}}$ to $\mathcal{A}$.

Decryption Key Reveal Query: Upon a query $(\mathsf{ID}, \mathsf{T}) \in \mathcal{I} \times \mathcal{T}$ from $\mathcal{A}$, $\mathcal{C}$ checks if the following conditions are simultaneously satisfied:

(a) $\mathsf{T} \leq \mathsf{T}_{\mathsf{cu}}$.

(b) $\mathsf{ID} \notin \mathsf{RL}_{\mathsf{T}}$.

(c) $(\mathsf{ID}, \mathsf{T}) \neq (\mathsf{ID}^*, \mathsf{T}^*)$.

(d) If $\mathsf{T}_{\mathsf{cu}} \geq \mathsf{T}^*$, $\mathsf{ID}^* \notin \mathsf{RL}_{\mathsf{T}^*}$, and $\mathsf{DK}_{\mathsf{ID}^*,\mathsf{T}}$ for the challenge ID^* has been revealed to $\mathcal{A}$ Q times via decryption key reveal queries $(\mathsf{ID}^*, \mathsf{T})$, then $\mathsf{ID} \neq \mathsf{ID}^*$.

(e) $(\mathsf{ID}, \mathsf{SK}_{\mathsf{ID}}) \in \mathtt{SKList}$.

[4] The condition (c) ensures the B-DKER security.

If these conditions are *not* satisfied, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ finds $\mathsf{SK}_{\mathsf{ID}}$ from $\mathtt{SKList}$, runs $\mathsf{DK}_{\mathsf{ID},\mathsf{T}} \leftarrow \mathsf{DKG}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{T}})$, and returns $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ to $\mathcal{A}$.

Challenge Query: $\mathcal{A}$ is allowed to make this query only once. Upon a query $M^* \in \mathcal{M}$ from $\mathcal{A}$, $\mathcal{C}$ picks the challenge bit $b \in \{0,1\}$ uniformly at random, runs $\mathsf{CT}^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ID}^*, \mathsf{T}^*, M^*)$ if $b = 0$ or samples CT^* uniformly at random from the ciphertext space if $b = 1$. Finally, $\mathcal{C}$ returns the challenge ciphertext CT^* to $\mathcal{A}$.

At some point, $\mathcal{A}$ outputs $b' \in \{0,1\}$ as its guess for b and terminates.
The above completes the description of the game.

Definition 3 (IND-sRID-Q-CPA)**.** *For any a-priori fixed Q ($:= poly(\lambda)$), we say that an RIBE scheme Π satisfies IND-sRID-Q-CPA security, if the advantage $\mathsf{Adv}^{IND\text{-}sQ\text{-}CPA}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot |\Pr[b' = b] - 1/2|$ is negligible for all PPT adversaries $\mathcal{A}$.*

The semi-adaptive security is a slight modification of the selective security; $\mathcal{A}$ sends the challenge identity/time period pair $(\mathsf{ID}^*, \mathsf{T}^*) \in \mathcal{I} \times \mathcal{T}$ to $\mathcal{C}$ just after $\mathcal{C}$ gives PP and KU_1 to $\mathcal{A}$.

The more desirable security notion, called *adaptive-identity* security, is defined in the same way as selective-identity security, except that in the security game the adversary $\mathcal{A}$ chooses the pair of the challenge identity and time period $(\mathsf{ID}^*, \mathsf{T}^*)$ not at the beginning of the game, but at the time it makes the challenge query. Since $\mathcal{C}$ does not know $(\mathsf{ID}^*, \mathsf{T}^*)$, the condition (b) in secret key reveal queries, the condition (b), (c) in revocation & key update queries, the condition (c) in decryption key reveal queries can be omitted in advance of the challenge query. In turn, the response to the challenge query is defined differently as follows:

Challenge Query: $\mathcal{A}$ is allowed to make this query only once. Upon a query $(\mathsf{ID}^*, \mathsf{T}^*, M^*)$ from $\mathcal{A}$, where it is required that the following conditions are satisfied simultaneously:

(a) if $\mathsf{SK}_{\mathsf{ID}^*}$ has been revealed to $\mathcal{A}$, then it is required that $\mathsf{ID}^* \in \mathsf{RL}_{\mathsf{T}^*}$,
(b) if $\mathsf{DK}_{\mathsf{ID}^*,\mathsf{T}}$ has been revealed to $\mathcal{A}$ more than Q times, then it is required that $\mathsf{ID}^* \in \mathsf{RL}_{\mathsf{T}^*}$,
(c) if $\mathsf{T}^* \leq \mathsf{T}_{\mathtt{cu}}$, then $\mathcal{A}$ has not submitted $(\mathsf{ID}^*, \mathsf{T}^*)$ as a decryption key reveal query,

$\mathcal{C}$ returns the challenge ciphertext CT^* in the same way as the selective-identity game.

Definition 4 (IND-RID-Q-CPA)**.** *For any a-priori fixed Q ($:= poly(\lambda)$), we say that an RIBE scheme Π satisfies IND-RID-Q-CPA security, if the advantage $\mathsf{Adv}^{IND\text{-}Q\text{-}CPA}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot |\Pr[b' = b] - 1/2|$ is negligible for all PPT adversaries $\mathcal{A}$.*

3.2 Treatment of Binary Trees

In the subsequent sections, we propose lattice-based and pairing-based B-DKER RIBE schemes. Since both constructions utilize the complete subtree (CS) method as several previous works, we explain the treatment of binary trees. Experts of R(H)IBE can skip this part.

KGC keeps a binary tree $\mathtt{BT}$ in the master secret key MK to manage users. The binary tree $\mathtt{BT}$ has N leaves and can manage at most N users. To manage arbitrary polynomial number of users, we set N super-polynomial in the security parameter λ, e.g., $N = \lambda^{\log(\lambda)}$. Let θ denote nodes of the binary tree $\mathtt{BT}$, and especially η denote leaf nodes. To realize the revocation mechanism, during secret key generations, KGC assigns each user ID to a randomly selected leaf node η in the binary tree $\mathtt{BT}$. In addition, during secret key generations and key update information generations, KGC assigns random elements, i.e., random vectors $\mathbf{u}_\theta \in \mathbb{Z}_q^n$ in the lattice-based scheme and random group elements P_θ
inG_2 in the pairing-based scheme, to some required nodes θ. When we say "the binary tree $\mathtt{BT}$", we assume that it contains the description of the nodes θ as natural numbers, assignment of users ID, and a random element in each node.

Although the binary tree $\mathtt{BT}$ has super-polynomially many leaves, the amount of information which KGC keeps is polynomial in the security parameter λ. To produce a secret key $\mathsf{SK}_{\mathsf{ID}}$, KGC keeps at most $O(\log N) = O(\log^2(\lambda))$ random elements in nodes $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})$. Since the number of secret key generation queries is polynomial in the security parameter λ, the amount of information produced during the queries is also polynomial. Similarly, to produce a key update KU_{T}, KGC keeps at most $O(R\log(N/R)) = O(R\log^2(\lambda))$ random elements in nodes $\theta \in \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$, where R denotes the number of revoked users. Since the life-time of the scheme is polynomial in the security parameter λ, the amount of information produced during key update queries is also polynomial.

To utilize the binary tree data structure and realize the revocation mechanism, we prepare the following four algorithms ($\mathsf{CS.SetUp}, \mathsf{CS.Assign}, \mathsf{CS.Cover}, \mathsf{CS.Match}$) in this paper:

$\mathsf{CS.SetUp}(N) \to \mathtt{BT}$: on input the natural number N, it outputs a binary tree $\mathtt{BT}$ with N leaves.

$\mathsf{CS.Assign}(\mathtt{BT}, \mathsf{ID}) \to (\eta, \mathtt{BT})$: on input a binary tree $\mathtt{BT}$ and an identity ID, it randomly assigns a leaf node η, which no other identities are still assigned to, to the identity ID. Then, it outputs the leaf η and also the "updated" binary tree $\mathtt{BT}$.

$\mathsf{CS.Cover}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}) \to \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$: on input a binary tree $\mathtt{BT}$ and a revocation list RL_{T}, it runs the KUNode algorithm and outputs a set of nodes $\mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$.

$\mathsf{CS.Match}(\mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}}), \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})) \to \theta$ or $\emptyset$: on input $\mathsf{Path}(\mathtt{BT}, \eta)$ and $\mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$, it outputs an arbitrary node $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}}) \cap \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$ if it exists. Otherwise, it outputs $\emptyset$.

3.3 Strategy-Dividing Lemma

As the previous works, during the security proof, we separate strategies of adversaries in the following two types:

- **Type-I adversary**: ID^* will be revoked before T^*. Hence, $\mathcal{A}$ may issue a secret key extraction query for $\mathsf{SK}_{\mathsf{ID}^*}$ or decryption key queries $\mathsf{DK}_{\mathsf{ID}^*,\mathsf{T}}$ for $\mathsf{T} \neq \mathsf{T}^*$ more than Q times.

- **Type-II adversary**: ID^* will not be revoked before T^*. Hence, $\mathcal{A}$ may issue decryption key queries $\mathsf{DK}_{\mathsf{ID}^*,\mathsf{T}}$ for $\mathsf{T} \neq \mathsf{T}^*$ at most Q times.

Let $\mathcal{A}_I$ and $\mathcal{A}_{II}$ denote adversaries which always follow the Type-I and the Type-II strategy, respectively. By definition, strategies of $\mathcal{A}_I$ and $\mathcal{A}_{II}$ cover *all possible strategies* of $\mathcal{A}$. Furthermore, the strategies are *publicly detectable*, i.e., during the security game, as soon as $\mathcal{A}$ deviates from the Type-i strategy, it can be efficiently recognized given $\mathcal{A}$'s view at the moment it deviates from the strategy. In general, $\mathcal{A}$ does not always follow either Type-I or Type-II, however, Katsumata et al. [KMT18] formally proved the following *strategy-dividing lemma*, i.e., if we can prove the security against each $\mathcal{A}_I$ and $\mathcal{A}_{II}$, we can prove it against general $\mathcal{A}$.

Lemma 7 (Strategy-Dividing Lemma [KMT18])**.** *Let Π be an RIBE scheme, and let $\mathcal{A}$ be any PPT adversary against the IND-sRID-Q-CPA security of Π. Let $\mathcal{A}_I$ and $\mathcal{A}_{II}$ be an adversary $\mathcal{A}$ which always follows the Type-I and the Type-II strategy, respectively. Let $\mathsf{Adv}^{\mathit{IND\text{-}sQ\text{-}CPA}}_{\Pi,\mathcal{A}_I}(\lambda)$ and $\mathsf{Adv}^{\mathit{IND\text{-}sQ\text{-}CPA}}_{\Pi,\mathcal{A}_{II}}(\lambda)$ denote $\mathcal{A}_I$'s and $\mathcal{A}_{II}$'s advantage against IND-sRID-Q-CPA security, respectively. Since two types cover all possible strategies and publicly detectable,*

$$\mathsf{Adv}^{\mathit{IND\text{-}sQ\text{-}CPA}}_{\Pi,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathit{IND\text{-}sQ\text{-}CPA}}_{\Pi,\mathcal{A}_I}(\lambda) + \mathsf{Adv}^{\mathit{IND\text{-}sQ\text{-}CPA}}_{\Pi,\mathcal{A}_{II}}(\lambda)$$

holds. The analogous inequality holds for an adversary against the IND-RID-Q-CPA security of Π.

4 B-DKER RIBE Scheme from Lattices

4.1 Construction

In this section, we construct a lattice-based B-DKER RIBE scheme by combining Chen et al.'s RIBE without DKER [CLL+12b] and CFFs.

We set the plaintext space as $\mathcal{M} = \{0,1\}$ and the identity space as $\mathcal{I} = \mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}$. We also encode the the time period space $\mathcal{T} = \{1, 2, \cdots, \mathsf{T}_{\max}\}$ into a polynomial sized subset of $\mathbb{Z}_q^n \setminus \{\mathbf{0}_n\}$. There are three matrices $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2$ in PP. We use the following hash functions to encode an identity and a time period into a matrix:

- $\mathbf{F}_{\mathsf{ID}} = \mathbf{A}_1 + H(\mathsf{ID})\mathbf{G} \in \mathbb{Z}_q^{n\times m}$,
- $\mathbf{F}_{\mathsf{T}} = \mathbf{A}_2 + H(\mathsf{T})\mathbf{G} \in \mathbb{Z}_q^{n\times m}$,

where $H(\cdot)$ is a FRD map defined in Definition 1 and $\mathbf{G}$ is a gadget matrix.

Our RIBE construction is as follows:

$\mathsf{SetUp}(1^n) \to (\mathsf{PP}, \mathsf{MK})$: Set parameters m, q, σ, α, and N. Run $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(q, n, m)$, $(w, d, \mathcal{F}) \leftarrow \mathsf{CFF.Gen}(|\mathcal{T}|, Q)$, and $\mathtt{BT} \leftarrow \mathsf{CS.SetUp}(N)$. Let $\mathsf{RL}_0 := \emptyset$. Sample $(\mathbf{A}_1, \mathbf{A}_2) \xleftarrow{\$} (\mathbb{Z}_q^{n\times m})^2$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$. Choose an FRD map $H(\cdot)$ as in Definition 1. Then, outputs

$$\mathsf{PP} := (H(\cdot), \mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{u}), \qquad \mathsf{MK} := (\mathtt{BT}, \mathsf{RL}_0, \mathbf{T}_{\mathbf{A}_0}).$$

$\mathsf{PKG}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID}) \to (\mathsf{SK}_{\mathsf{ID}}, \mathsf{MK}')$: Run $(\eta_{\mathsf{ID}}, \mathtt{BT}') \leftarrow \mathsf{CS.Assign}(\mathtt{BT}, \mathsf{ID})$. For each node $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})$, recall $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \in (\mathbb{Z}_q^n)^d$ if they were defined. Otherwise, choose random $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \xleftarrow{\$} (\mathbb{Z}_q^n)^d$ and store them in θ. For every $\ell \in [d]$, sample $\mathbf{e}'_{\theta,\ell} \leftarrow \mathsf{SampleLeft}(\mathbf{A}_0, \mathbf{F}_{\mathsf{ID}}, \mathbf{u}'_{\theta,\ell}, \mathbf{T}_{\mathbf{A}_0}, \sigma)$. Then, it outputs

$$\mathsf{SK}_{\mathsf{ID}} = \left(\left\{\theta, \{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}\right\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}\right), \qquad \mathsf{MK}' := (\mathtt{BT}', \mathsf{RL}_{\mathsf{T}}, \mathbf{T}_{\mathbf{A}_0}).$$

$\mathsf{KeyUp}(\mathsf{PP}, \mathsf{T}, \mathsf{MK}) \to (\mathsf{KU}_{\mathsf{T}}, \mathsf{MK}')$: Run $\mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}) \leftarrow \mathsf{CS.Cover}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$. For each node $\theta \in \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$, recall $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \in (\mathbb{Z}_q^n)^d$ if they were defined. Otherwise, choose random $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \xleftarrow{\$} (\mathbb{Z}_q^n)^d$ and store them in θ. Sample $\tilde{\mathbf{e}}_\theta \leftarrow \mathsf{SampleLeft}(\mathbf{A}_0, \mathbf{F}_{\mathsf{T}}, \tilde{\mathbf{u}}_\theta, \mathbf{T}_{\mathbf{A}_0}, \sigma)$. Then, it outputs

$$\mathsf{KU}_{\mathsf{T}} = \left(\{\theta, \tilde{\mathbf{e}}_\theta\}_{\theta\in\mathsf{KUNode}(\mathtt{BT},\mathsf{RL}_{\mathsf{T}})}, \mathcal{F}_{\mathsf{T}}\right), \qquad \mathsf{MK}' := (\mathtt{BT}', \mathsf{RL}_{\mathsf{T}}, \mathbf{T}_{\mathbf{A}_0}).$$

$\mathsf{DKG}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{T}}) \to \mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ or $\perp$: Parse $\mathsf{SK}_{\mathsf{ID}}$ and KU_{T} as $\left(\left\{\theta, \{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}\right\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}\right)$ and $\left(\{\theta, \tilde{\mathbf{e}}_\theta\}_{\theta\in\mathsf{KUNode}(\mathtt{BT},\mathsf{RL}_{\mathsf{T}})}, \mathcal{F}_{\mathsf{T}}\right)$, respectively. Output $\perp$ if $\emptyset \leftarrow \mathsf{CS.Match}(\mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}}), \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}))$. Otherwise, for $\theta \leftarrow \mathsf{CS.Match}(\mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}}), \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}))$, compute $\mathbf{e}_\theta = \sum_{\ell\in\mathcal{F}_{\mathsf{T}}} \mathbf{e}'_{\theta,\ell}$ and output

$$\mathsf{DK}_{\mathsf{ID},\mathsf{T}} = (\mathbf{e}_\theta, \tilde{\mathbf{e}}_\theta).$$

$\mathsf{Enc}(\mathsf{PP}, \mathsf{ID}, \mathsf{T}, M) \to \mathsf{CT}_{\mathsf{ID},\mathsf{T}}$: To encrypt a bit $M \in \{0,1\}$, it samples a secret vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, random matrices $\left(\mathbf{R}_{\mathsf{ID}}, \mathbf{R}_{\mathsf{T}}\right) \xleftarrow{\$} \left(\{-1,1\}^{m\times m}\right)^2$, noise $x \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, and a noise vector $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$. It computes

$$c_0 = \mathbf{u}^\top\mathbf{s} + x + M\left\lfloor\frac{q}{2}\right\rfloor \in \mathbb{Z}_q, \qquad \mathbf{c} = \left[\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}}|\mathbf{F}_{\mathsf{T}}\right]^\top\mathbf{s} + \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{ID}}^\top \\ \mathbf{R}_{\mathsf{T}}^\top \end{bmatrix}\mathbf{y} \in \mathbb{Z}_q^{3m},$$

and outputs a ciphertext $\mathsf{CT}_{\mathsf{ID},\mathsf{T}} := (c_0, \mathbf{c}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{3m}$.

$\mathsf{Dec}(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{T}}, \mathsf{CT}) \to M$: Parse $\mathbf{c}$ as $[\mathbf{c}_0|\mathbf{c}_1|\mathbf{c}_2]^\top$, where $\mathbf{c}_i \in \mathbb{Z}_q^m$. Compute

$$c' = c_0 - \mathbf{e}_\theta^\top\begin{bmatrix}\mathbf{c}_0\\ \mathbf{c}_1\end{bmatrix} - \tilde{\mathbf{e}}_\theta^\top\begin{bmatrix}\mathbf{c}_0\\ \mathbf{c}_2\end{bmatrix} \in \mathbb{Z}_q.$$

Compare c' and $\lfloor\frac{q}{2}\rfloor$ treating them as integers in $\mathbb{Z}$. If they are close, i.e., if $|c' - \lfloor\frac{q}{2}\rfloor| < \lfloor\frac{q}{4}\rfloor$, output 1, otherwise output 0.

Parameters and Correctness. Due to the property of the CS method, an output of $\mathsf{CS.Match}(\mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}}), \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}))$ is not $\perp$ for non-revoked ID. We have during decryption,

$$c' = c_0 - \mathbf{e}_\theta^\top\begin{bmatrix}\mathbf{c}_0\\ \mathbf{c}_1\end{bmatrix} - \tilde{\mathbf{e}}_\theta^\top\begin{bmatrix}\mathbf{c}_0\\ \mathbf{c}_2\end{bmatrix}$$

$$= M \left\lfloor \frac{q}{2} \right\rfloor + \underbrace{x - \mathbf{e}_\theta^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{ID}}^\top \end{bmatrix} \mathbf{y} - \tilde{\mathbf{e}}_\theta^\top \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{T}}^\top \end{bmatrix} \mathbf{y}}_{\text{error term}}.$$

Let $\mathbf{e}_\theta = [\mathbf{e}_{\theta,1} | \mathbf{e}_{\theta,2}]$ and $\tilde{\mathbf{e}}_\theta = [\tilde{\mathbf{e}}_{\theta,1} | \tilde{\mathbf{e}}_{\theta,2}]$ with $[\mathbf{e}_{\theta,1}, \mathbf{e}_{\theta,2}, \tilde{\mathbf{e}}_{\theta,1}, \tilde{\mathbf{e}}_{\theta,2}] \in (\mathbb{Z}^m)^4$. Then the error term is

$$x - \left(\mathbf{e}_{\theta,1} + \tilde{\mathbf{e}}_{\theta,1} + \mathbf{R}_{\mathsf{ID}} \mathbf{e}_{\theta,2} + \mathbf{R}_{\mathsf{T}} \tilde{\mathbf{e}}_{\theta,2}\right)^\top \mathbf{y}.$$

From Lemma 1, we have $\|\mathbf{e}'_{\theta,\ell}\| \leq \sigma\sqrt{2m}$, $\|\tilde{\mathbf{e}}_\theta\| \leq \sigma\sqrt{2m}$, $|x| \leq \alpha q$, and $\|\mathbf{y}\| \leq \alpha q \sqrt{m}$ with high probability. Since the underlying CFF is w-uniform, we have

$$\|\mathbf{e}_\theta\| \leq \sum_{\ell \in \mathcal{F}_{\mathsf{T}}} \|\mathbf{e}'_{\theta,\ell}\| \leq w\sigma\sqrt{2m}.$$

From Lemma 4, $\|\mathbf{R}_{\mathsf{ID}}\| \leq O(\sqrt{m})$ and $\|\mathbf{R}_{\mathsf{T}}\| \leq O(\sqrt{m})$ with high probability. Then,

$$\|\mathbf{e}_{\theta,1} + \tilde{\mathbf{e}}_{\theta,1} + \mathbf{R}_{\mathsf{ID}} \mathbf{e}_{\theta,2} + \mathbf{R}_{\mathsf{T}} \tilde{\mathbf{e}}_{\theta,2}\| \leq \|\mathbf{e}_\theta\| + \|\tilde{\mathbf{e}}_\theta\| + \|\mathbf{R}_{\mathsf{ID}}\| \cdot \|\mathbf{e}_\theta\| + \|\mathbf{R}_2\| \cdot \|\tilde{\mathbf{e}}_\theta\| \leq O(w\sigma m).$$

Thus, the error term is bounded above by

$$|x| + \left| [\mathbf{e}_{\theta,1} + \tilde{\mathbf{e}}_{\theta,1} + \mathbf{R}_{\mathsf{ID}} \mathbf{e}_{\theta,2} + \mathbf{R}_2 \tilde{\mathbf{e}}_{\theta,2}]^\top \mathbf{y} \right| \leq w\sigma m^{3/2} q\alpha$$

with high probability.

Now, for the scheme to work correctly, the following conditions should hold, taking n to be the security parameter:

- the error term is less than $q/5$ with high probability, i.e., $\alpha < (w\sigma m^{3/2})^{-1}$,
- that $\mathsf{TrapGen}$ can operate, i.e., $m > 2n \log q$,
- that σ is sufficiently large for $\mathsf{SampleLeft}$ and $\mathsf{SampleRight}$, i.e., $\sigma > \|\mathbf{T}_{\mathbf{G}}\|_{\mathsf{GS}} \cdot \|\mathbf{R}_{\mathsf{ID}}\| \cdot \omega(\sqrt{\log m}) = \sqrt{m} \cdot \omega(\sqrt{\log m})$,
- that Regev's reduction applies, i.e., $q > 2\sqrt{n}/\alpha$,

Hence, we set the parameters as follows:

$$m = 2n^{1+\delta},\ q = wm^{5/2} \cdot \omega(\sqrt{\log n}),\ \sigma = \sqrt{m} \cdot \omega(\sqrt{\log n}),\ \alpha = \left(wm^2 \cdot \omega(\sqrt{\log n})\right)^{-1}.$$

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

4.2 Security

In this section, we prove the security of our scheme in Section 4.1.

Theorem 1. *If the LWE assumption holds and the underlying CFF is w-uniform Q-cover-free, then the proposed RIBE scheme in Section 4.1 with the parameters set as above is IND-sRID-Q-CPA secure.*

The proof proceeds in a sequence of games, where the first game is the same as IND-sRID-Q-CPA game. In the last game, the challenge ciphertext is a uniform random element in the ciphertext space, hence, the advantage of a PPT adversary $\mathcal{A}$ is zero. Let E_i denote the event that $\mathcal{A}$ wins the game, i.e., $b' = b$, in **Game** i. Then, $\mathcal{A}$'s advantage in **Game** i is $\left|\Pr[E_i] - \frac{1}{2}\right|$. As we claimed in Section 3.3, we provide distinct reductions against the Type-I and the Type-II adversary. The proof against the Type-I adversary is similar to that of Chen et al.[CLL+12b]. During the proof against the Type-II adversary, we use the property of CFFs and answer Q bounded decryption key queries on $(\mathsf{ID}^*, \mathsf{T})$.

Proof. The proof proceeds in a sequence of games.

Game^{real}: This is the original IND-sRID-Q-CPA game between an adversary $\mathcal{A}$ against our scheme and an IND-sRID-Q-CPA challenger $\mathcal{C}$.

Game 0: This game is a preparation to utilize the property of CFFs later. We separate descriptions of **Game** 0 depending on types of the adversary.

Type-I Adversary: The game is the same as **Game**^{real}. Thus,

$$\Pr[E_0] = \Pr[E_{real}].$$

Type-II Adversary: Let $\{\mathsf{T}_i\}_{i\in[Q]}$ be a set of time periods, where $\mathcal{A}$ issues decryption key reveal queries $(\mathsf{ID}^*, \mathsf{T}_i)$. Notice that $\mathcal{C}$ does not know the actual values of $\{\mathsf{T}_i\}_{i\in[Q]}$ at the beginning of the game.

The game is the same as **Game**^{real} except that at the beginning of the game, $\mathcal{C}$ guesses an index $\ell^* \in \mathcal{F}_{\mathsf{T}^*}$ such that the short vector $\mathbf{e}'_{\theta,\ell^*}$ is not used to answer $\mathcal{A}$'s decryption key reveal queries on $(\mathsf{ID}^*, \mathsf{T}_i)$, i.e.,

$$\ell^* \in \mathcal{F}_{\mathsf{T}^*} \ \wedge \ \ell^* \notin \mathcal{F}_{\mathsf{T}_1} \ \wedge \ \cdots \ \wedge \ \ell^* \notin \mathcal{F}_{\mathsf{T}_Q}. \tag{2}$$

If the guess is incorrect, $\mathcal{C}$ aborts the game and output a random bit.

If $\mathcal{C}$ does not abort the game, the game is the same as **Game**^{real} from $\mathcal{A}$'s view. Since the underlying CFF is w-uniform, $\mathcal{C}$'s guess is correct with probability $1/w$. Thus,

$$\left|\Pr[E_0] - \frac{1}{2}\right| = \frac{1}{w}\left|\Pr[E_{real}] - \frac{1}{2}\right|.$$

In the rest of the proof, $\mathcal{C}$ utilizes the knowledge of the index ℓ^*.

Game 1: In **Game** 0, PP contains random matrices $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2$ in $\mathbb{Z}_q^{n\times m}$. At the challenge phase, $\mathcal{C}$ creates the challenge ciphertext $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$. We use $\mathbf{R}_{\mathsf{ID}^*}$ and $\mathbf{R}_{\mathsf{T}^*}$ to denote random matrices which $\mathcal{C}$ will use to create the challenge ciphertext.

A modification in this game is the same as that of Chen et al. [CLL+12b]. **Game** 1 is the same as **Game** 0 except that $\mathcal{C}$ changes the way to create random matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ in PP. At the setup phase, $\mathcal{C}$ samples random matrices $\mathbf{R}_{\mathsf{ID}^*}$ and $\mathbf{R}_{\mathsf{T}^*}$, which will be used to create the challenge ciphertext $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$, and creates public matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ as

$$\mathbf{A}_1 \leftarrow \mathbf{A}_0\mathbf{R}_{\mathsf{ID}^*} - H(\mathsf{ID}^*)\mathbf{G} \qquad \text{and} \qquad \mathbf{A}_2 \leftarrow \mathbf{A}_0\mathbf{R}_{\mathsf{T}^*} - H(\mathsf{T}^*)\mathbf{G}. \tag{3}$$

The remainder of the game is unchanged.

We show that **Game** 0 and **Game** 1 are statistically close in $\mathcal{A}$'s view. The only difference between the games is whether the matrices $\mathbf{R}_{\mathsf{ID}^*}$ and $\mathbf{R}_{\mathsf{T}^*}$ are used to create $\mathbf{A}_1$ and $\mathbf{A}_2$ in PP or not. Let $\mathbf{R}^* := [\mathbf{R}_{\mathsf{ID}^*}|\mathbf{R}_{\mathsf{T}^*}] \in \mathbb{Z}_q^{m\times 2m}$. From Lemma 5, the following two distributions are statistically close:

$$\left(\mathbf{A}_0, [\mathbf{A}_1|\mathbf{A}_2], (\mathbf{R}^*)^\top \mathbf{y}\right) \quad \text{and} \quad \left(\mathbf{A}_0, \mathbf{A}_0\mathbf{R}^*, (\mathbf{R}^*)^\top \mathbf{y}\right)$$

for independently random matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ in $\mathbb{Z}_q^{n\times m}$. It follows that $\mathbf{A}_0\mathbf{R}_{\mathsf{ID}^*}$ and $\mathbf{A}_0\mathbf{R}_{\mathsf{T}^*}$ are independently random matrices in $\mathbb{Z}_q^{n\times m}$. Hence, **Game** 0 and **Game** 1 are statistically close in $\mathcal{A}$'s view. Thus,

$$\left|\Pr[E_1] - \frac{1}{2}\right| \le \left|\Pr[E_0] - \frac{1}{2}\right| - \mathsf{negl}(\lambda).$$

Game 2: In **Game** 1, $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]}$ are independently random vectors in $\mathbb{Z}_q^n$, and $\mathcal{C}$ samples short vectors $\{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}$ and $\tilde{\mathbf{e}}_\theta$ using the SampleLeft algorithm. **Game** 2 is the same as **Game** 1 except the assignment of ID^* in BT and creations of

- vectors $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]}$ in BT,
- short vectors $\{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}$ for ID^*,
- short vectors $\tilde{\mathbf{e}}_\theta$ for T^*,

so that $\mathcal{C}$ creates the keys without using the trapdoor $\mathbf{T}_{\mathbf{A}_0}$.

At first, we explain the assignment of ID^* in BT. Just after $\mathcal{C}$ generates BT, $\mathcal{C}$ picks a random leaf node in BT and set it as η^*. Upon $\mathcal{A}$'s secret key generation queries for $\mathsf{ID} \neq \mathsf{ID}^*$, $\mathcal{C}$ picks a random leaf node η_{ID} in $\mathtt{BT} \setminus \{\eta^*\}$ and assign ID. Upon $\mathcal{A}$'s secret key generation query for ID^*, $\mathcal{C}$ assigns ID^* to η^*. The modification does not change the distribution of the game.

Then, we separate the way for creating the above vectors depending on types of the adversary. Although $\mathcal{C}$ first samples $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]}$ then samples $\{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}$ and $\tilde{\mathbf{e}}_\theta$ using the SampleLeft algorithm in **Game** 1, $\mathcal{C}$ first samples $\{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}$ for ID^* and $\tilde{\mathbf{e}}_\theta$ for T^* using the $\mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ algorithm then uses them to create $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]}$ in **Game** 2.

Type-I Adversary: A modification in this game is similar to that of Chen et al. [CLL+12b]. By definition, short vectors $\{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}$ used to create $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{DK}_{\mathsf{ID}^*,\mathsf{T}}$ are associated with nodes

$\theta \in \mathsf{Path}(\mathsf{BT}, \eta^*)$. By definition of the Type-I adversary, since ID^* will be revoked by T^*, short vectors $\tilde{\mathbf{e}}_\theta$ in $\mathsf{KU}_{\mathsf{T}^*}$ have to be associated with nodes $\theta \notin \mathsf{Path}(\mathsf{BT}, \eta^*)$ due to the property of the CS method. Observe that there are no nodes θ that are common for $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$. Since $\mathcal{C}$ has already set the leaf node η^*, we change the creations of the above vectors as follows:

- For $\theta \in \mathsf{Path}(\mathsf{BT}, \eta^*)$, $\mathcal{C}$ samples an independently random vector $\mathbf{e}'_{\theta,\ell} \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell} = [\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}^*}]\mathbf{e}_{\theta,\ell}$ for $\ell \in [d]$.
- For $\theta \notin \mathsf{Path}(\mathsf{BT}, \eta^*)$, the creation of $\mathbf{u}'_{\theta,\ell}$ is the same as that of **Game** 1 for $\ell \in [d] \setminus \mathcal{F}_{\mathsf{T}^*}$. For the nodes, $\mathcal{C}$ samples $\tilde{\mathbf{e}}_\theta \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell}$ for $\ell \in \mathcal{F}_{\mathsf{T}^*}$ as uniformly random vectors in $\mathbb{Z}_q^n$ subject to $\mathbf{u} - \sum_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \mathbf{u}'_{\theta,\ell} = [\mathbf{A}_0|\mathbf{F}_{\mathsf{T}^*}]\tilde{\mathbf{e}}_\theta$.

Then, $\mathcal{C}$ creates $\mathsf{SK}_{\mathsf{ID}}$, KU_{T}, and $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ to answer $\mathcal{A}$'s queries as follows:

- $\mathcal{C}$'s creations of $\mathsf{SK}_{\mathsf{ID}}$ for $\mathsf{ID} \neq \mathsf{ID}^*$ and KU_{T} for $\mathsf{T} \neq \mathsf{T}^*$ are unchanged,
- $\mathcal{C}$ uses the above $\{\mathbf{e}'_{\theta,\ell}\}_{\ell \in [d]}$ as $\mathsf{SK}_{\mathsf{ID}^*}$,
- $\mathcal{C}$ uses the above $\tilde{\mathbf{e}}_\theta$ as $\mathsf{KU}_{\mathsf{T}^*}$.
- $\mathcal{C}$ uses the above $\mathsf{SK}_{\mathsf{ID}}$ and KU_{T} to create $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$.

Here, $\mathcal{C}$ does not use the trapdoor $\mathbf{T}_{\mathbf{A}_0}$ to create $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$.

We show that **Game** 2 and **Game** 1 are statistically close with high probability in $\mathcal{A}$'s view, where the discussion is the same as that of Chen et al. [CLL+12b]. In **Game** 1, observe that

- vectors $\mathbf{u}'_{\theta,\ell}$ distribute according to uniform in $\mathbb{Z}_q^n$,
- short vectors $\mathbf{e}'_{\theta,\ell}$ of $\mathsf{SK}_{\mathsf{ID}^*}$ distribute according to $\mathcal{D}_{\Lambda_q^{\mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}^*}]),\sigma}$,
- short vectors $\tilde{\mathbf{e}}_\theta$ for T^* distribute according to $\mathcal{D}_{\Lambda_q^{\mathbf{u}-\sum_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{T}^*}]),\sigma}$.

In **Game** 2, $\{\mathbf{e}'_{\theta,\ell}\}_{\ell \in [d]}$ for ID^* and $\tilde{\mathbf{e}}_\theta$ for T^* distribute according to $\mathcal{D}_{\mathbb{Z}^{2m},\sigma}$ due to the property of $\mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$. Hence, from Lemma 2, the distribution of each $\{\mathbf{u}'_{\theta,\ell}\}_{\ell \in [d]}$ is statistically close to uniform in $\mathbb{Z}_q^n$ as **Game** 1. Furthermore, the conditional distribution of each $\{\mathbf{e}'_{\theta,\ell}\}_{\ell \in [d]}$ and $\tilde{\mathbf{e}}_\theta$ given $\{\mathbf{u}'_{\theta,\ell}\}_{\ell \in [d]}$ is $\mathcal{D}_{\Lambda_q^{\mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}^*}]),\sigma}$ and $\mathcal{D}_{\Lambda_q^{\mathbf{u}-\sum_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{T}^*}]),\sigma}$, respectively, as **Game** 1. Hence, **Game** 2 and **Game** 1 are statistically close with high probability in $\mathcal{A}$'s view. Thus,

$$\left|\Pr[E_2] - \frac{1}{2}\right| \leq \left|\Pr[E_1] - \frac{1}{2}\right| - \mathsf{negl}(\lambda).$$

Type-II adversary. A modification of **Game** 2 is the most technical part of this paper. By definition of the Type-II adversary, since ID^* will not be revoked by T^*, short vectors $\tilde{\mathbf{e}}_\theta$ in $\mathsf{KU}_{\mathsf{T}^*}$ have to be associated with nodes $\theta \in \mathsf{Path}(\mathsf{BT}, \eta^*)$ due to the property of the CS method. Observe that there are nodes θ that are common for $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$. Since $\mathcal{C}$ has already set the leaf node η^*, we change the creations of the above vectors as follows:

- For $\theta \in \mathsf{Path}(\mathtt{BT}, \eta^*)$, $\mathcal{C}$ samples an independently random vector $\mathbf{e}'_{\theta,\ell} \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell} = [\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}^*}]\mathbf{e}_{\theta,\ell}$ for $\ell \in [d] \setminus \{\ell^*\}$. For the nodes, $\mathcal{C}$ samples $\tilde{\mathbf{e}}_\theta \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell} = \mathbf{u} - \sum_{\ell \in \mathcal{F}_{\mathsf{T}^*} \setminus \{\ell^*\}} \mathbf{u}'_{\theta,\ell} - [\mathbf{A}_0|\mathbf{F}_{\mathsf{T}^*}]\tilde{\mathbf{e}}_\theta$.
- For $\theta \notin \mathsf{Path}(\mathtt{BT}, \eta^*)$, the creations of $\mathbf{u}'_{\theta,\ell}$ and $\tilde{\mathbf{e}}_\theta$ are the same as in **Game** 2 against the type I adversary.

Then, $\mathcal{C}$ creates $\mathsf{SK}_{\mathsf{ID}}$, KU_{T}, and $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ to answer $\mathcal{A}$'s queries as follows:

- $\mathcal{C}$'s creations of $\mathsf{SK}_{\mathsf{ID}}$ for $\mathsf{ID} \neq \mathsf{ID}^*$ and KU_{T} for $\mathsf{T} \neq \mathsf{T}^*$ are unchanged,
- $\mathcal{C}$ uses the above $\{\mathbf{e}'_{\theta,\ell}\}_{\ell \in [d]}$ as $\mathsf{SK}_{\mathsf{ID}^*}$,
- $\mathcal{C}$ uses the above $\tilde{\mathbf{e}}_\theta$ as $\mathsf{KU}_{\mathsf{T}^*}$.
- $\mathcal{C}$ uses the above $\mathsf{SK}_{\mathsf{ID}}$ and KU_{T} to create $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$.

Here, $\mathcal{C}$ does not use the trapdoor $\mathbf{T}_{\mathbf{A}_0}$ to create $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$.

We show that **Game** 2 and **Game** 1 are statistically close with high probability in $\mathcal{A}$'s view. At first, we observe that $\mathcal{C}$ is able to answer all key queries by $\mathcal{A}$. In particular, $\mathcal{C}$ is able to answer $\mathsf{DK}_{\mathsf{ID}^*,\mathsf{T}}$ queries although it does not have $\mathbf{e}'_{\theta,\ell^*}$. In **Game** 0, we set the index ℓ^* so that $\mathbf{e}'_{\theta,\ell^*}$ is not used to answer decryption key reveal queries as stated in (2). Hence, the absence of $\mathbf{e}'_{\theta,\ell^*}$ does not occur any problems. Then, the remaining proof is the same as that of **Game** 2 against the Type-I adversary. All vectors $\mathbf{u}'_{\theta,\ell}$, $\mathbf{e}'_{\theta,\ell}$, and $\tilde{\mathbf{e}}_\theta$ distribute the same way as in **Game 1** with high probability in $\mathcal{A}$'s view from Lemma 2. Thus,

$$\left|\Pr[E_2] - \frac{1}{2}\right| \leq \left|\Pr[E_1] - \frac{1}{2}\right| - \mathsf{negl}(\lambda).$$

Game 3: In **Game** 2, a matrix $\mathbf{A}_0$ is generated by the $\mathsf{TrapGen}$ algorithm and its trapdoor $\mathbf{T}_{\mathbf{A}_0}$ is used to create $\mathsf{SK}_{\mathsf{ID}}$ for $\mathsf{ID} \neq \mathsf{ID}^*$ and KU_{T} for $\mathsf{T} \neq \mathsf{T}^*$. **Game** 3 is the same as **Game** 2 except that we sample $\mathbf{A}_0$ as a random matrix in $\mathbb{Z}_q^{n \times m}$. From the property of the $\mathsf{TrapGen}$ algorithm, the matrix $\mathbf{A}_0$ generated by the algorithm is statistically close to a uniformly random matrix in $\mathbb{Z}_q^{n \times m}$. Hence, the distributions of PP are statistically close between **Game** 2 and **Game** 3.

Then, we show they way $\mathcal{C}$ creates $\mathsf{SK}_{\mathsf{ID}}$ for $\mathsf{ID} \neq \mathsf{ID}^*$ and KU_{T} for $\mathsf{T} \neq \mathsf{T}^*$ without using the trapdoor $\mathbf{T}_{\mathbf{A}_0}$. Observe that

$$[\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}}] := [\mathbf{A}_0|\mathbf{A}_1 + H(\mathsf{ID})\mathbf{G}] = [\mathbf{A}_0|\mathbf{A}_0\mathbf{R}_{\mathsf{ID}^*} + (H(\mathsf{ID}) - H(\mathsf{ID}^*))\,\mathbf{G}],$$
$$[\mathbf{A}_0|\mathbf{F}_{\mathsf{T}}] := [\mathbf{A}_0|\mathbf{A}_2 + H(\mathsf{T})\mathbf{G}] = [\mathbf{A}_0|\mathbf{A}_0\mathbf{R}_{\mathsf{T}^*} + (H(\mathsf{T}) - H(\mathsf{T}^*))\,\mathbf{G}].$$

Due to the property of gadget matrix $\mathbf{G}$, we know a trapdoor $\mathbf{T}_{\mathbf{G}}$ which is also a trapdoor for $(H(\mathsf{ID}) - H(\mathsf{ID}^*))\mathbf{G}$ and $(H(\mathsf{T}) - H(\mathsf{T}^*))\mathbf{G}$ if $\mathsf{ID} \neq \mathsf{ID}^*$ and $\mathsf{T} \neq \mathsf{T}^*$, since $H(\mathsf{ID}) - H(\mathsf{ID}^*)$ and $H(\mathsf{T}) - H(\mathsf{T}^*)$ in $\mathbb{Z}_q^{n \times n}$ are full rank. Furthermore, $\mathcal{C}$ also knows random matrices $\mathbf{R}_{\mathsf{ID}^*}$ and $\mathbf{R}_{\mathsf{T}^*}$. Then, $\mathcal{C}$ creates $\{\mathbf{e}'_{\theta,\ell}\}_{\ell \in [d]} \in \mathsf{SK}_{\mathsf{ID}}$ for $\mathsf{ID} \neq \mathsf{ID}^*$ and $\tilde{\mathbf{e}}_\theta \in \mathsf{KU}_{\mathsf{T}}$ for $\mathsf{T} \neq \mathsf{T}^*$ by using the $\mathsf{SampleRight}$ algorithm, i.e.,

- $\mathbf{e}'_{\theta,\ell} \leftarrow \mathsf{SampleRight}(\mathbf{A}_0, H(\mathsf{ID})\mathbf{G}, \mathbf{R}_{\mathsf{ID}^*}, \mathbf{u}'_{\theta,\ell}, \mathbf{T}_{\mathbf{G}}, \sigma)$,
- $\tilde{\mathbf{e}}_\theta \leftarrow \mathsf{SampleRight}(\mathbf{A}_0, H(\mathsf{T})\mathbf{G}, \mathbf{R}_{\mathsf{T}^*}, \mathbf{u} - \sum_{\ell \in \mathcal{F}_{\mathsf{T}}} \mathbf{u}'_{\theta,\ell}, \mathbf{T}_{\mathbf{G}}, \sigma)$.

The distributions of the vectors are statistically close between **Game** 2 and **Game** 3. In **Game** 2, the distributions of $\mathbf{e}'_{\theta,\ell}$ and $\tilde{\mathbf{e}}_\theta$ are statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}}]),\sigma}$ and $\mathcal{D}_{\Lambda_q^{\mathbf{u}-\sum_{\ell\in\mathcal{F}_{\mathsf{T}}}\mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{T}}]),\sigma}$ due to the property of $\mathsf{SampleLeft}$. In **Game** 3, the distributions of $\mathbf{e}'_{\theta,\ell}$ and $\tilde{\mathbf{e}}_\theta$ are statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}}]),\sigma}$ and $\mathcal{D}_{\Lambda_q^{\mathbf{u}-\sum_{\ell\in\mathcal{F}_{\mathsf{T}}}\mathbf{u}'_{\theta,\ell}}([\mathbf{A}_0|\mathbf{F}_{\mathsf{T}}]),\sigma}$ due to the property of $\mathsf{SampleRight}$. Hence, **Game** 3 and **Game** 2 are statistically close in $\mathcal{A}$'s view. Thus,

$$\left|\Pr[E_3] - \frac{1}{2}\right| \leq \left|\Pr[E_2] - \frac{1}{2}\right| - \mathsf{negl}(\lambda).$$

Game^{final}: **Game**^{final} is the same as **Game** 3 except that the challenge ciphertext $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*} = (c_0, \mathbf{c})$ is always chosen as a random independent element in the ciphertext space $\mathbb{Z}_q \times \mathbb{Z}_q^{3m}$. Since the challenge ciphertext is always a fresh random element in the ciphertext space, $\mathcal{A}$'s advantage in this game is zero.

We show that **Game** 3 and **Game**^{final} are computationally indistinguishable for a PPT adversary under the LWE assumption. Thus,

$$\left|\Pr[E_3] - \frac{1}{2}\right| = |\Pr[E_3] - \Pr[E_{final}]| \leq \mathsf{Adv}_{\mathcal{B}}^{LWE}.$$

Reduction from LWE. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing **Game** 3 and **Game**^{final}. Then we use $\mathcal{A}$ to construct a PPT algorithm $\mathcal{B}$ which solves LWE with non-negligible advantage.

By definition, an LWE problem instance $(\mathbf{A}, \mathbf{v})$ consists of a uniformly random matrix $\mathbf{A}$ and a vector $\mathbf{v}$ is either noisy inner-product with a secret vector $\mathbf{s}$; $\mathbf{v} = \mathbf{A}^\top\mathbf{s}+\mathbf{x}$, or a truly random vector $\mathbf{v}$. Given the former distribution, $\mathcal{B}$ can create a challenge ciphertext distributed as in **Game** 3 whereas $\mathcal{B}$ can create a challenge ciphertext distributed as in **Game**^{final} otherwise. $\mathcal{B}$ uses the adversary $\mathcal{A}$ to distinguish between the two distributions. $\mathcal{B}$ proceeds as follows:

Instance: $\mathcal{B}$ is given the LWE instance; a random matrix $[\mathbf{u}|\mathbf{A}_0] \in \mathbb{Z}_q^{n\times(m+1)}$ along with a vector $\mathbf{v} = [v_0|\mathbf{v}'] \in \mathbb{Z}_q^{m+1}$.

Targeting: $\mathcal{A}$ announces to $\mathcal{B}$ the target identity ID^* and the target time period T^* that intends to attack.

SetUp: $\mathcal{B}$ sets $(\mathbf{A}_0, \mathbf{u})$ as the LWE instance. $\mathcal{B}$ samples $\mathbf{R}_{\mathsf{ID}^*}, \mathbf{R}_{\mathsf{T}^*}$ and computes $\mathbf{A}_1, \mathbf{A}_2$ as (3) in **Game** 1. $\mathcal{B}$ runs $(w, d, \mathcal{F}) \xleftarrow{\$} \mathsf{CFF.Gen}(|\mathcal{T}|, Q)$ and guesses an index $\ell^* \in \mathcal{F}_{\mathsf{T}^*}$ for Type-II adversary $\mathcal{A}$ as stated in **Game** 0. $\mathcal{B}$ runs $\mathtt{BT} \leftarrow \mathsf{CS.SetUp}(N)$ and picks a random leaf node η^* as stated in **Game** 2. Let $\mathsf{RL}_0 = \emptyset$. $\mathcal{B}$ chooses an FRD map $H(\cdot)$ as in Definition 1. Then, $\mathcal{B}$ outputs $\mathsf{PP} := (H(\cdot), \mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{u})$ and $\mathsf{MK} = (\mathtt{BT}, \ell^*, \mathsf{RL}_0, \mathbf{R}_{\mathsf{ID}^*}, \mathbf{R}_{\mathsf{T}^*})$.

Secret Key Generation Query: $\mathcal{B}$ is given a query $\mathsf{ID} \in \mathcal{I}$ from $\mathcal{A}$, where it is required that $(\mathsf{ID}, *) \notin \mathtt{SKList}$. If $\mathsf{ID} \neq \mathsf{ID}^*$, $\mathcal{B}$ pick an unassigned leaf node uniformly random in $\mathtt{BT} \setminus \{\eta^*\}$ and assigns ID to the node, while $\mathcal{B}$ assigns ID^* to η^* as stated in **Game** 2. For each node $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})$, recall $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \in (\mathbb{Z}_q^n)^d$ if they were defined. Otherwise, $\mathcal{B}$ creates random vectors $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \xleftarrow{\$} (\mathbb{Z}_q^n)^d$ as stated in **Game 2** and store them in θ:
When $\mathcal{A}$ is type-I:

- For $\theta \in \mathsf{Path}(\mathtt{BT}, \eta^*)$, $\mathcal{C}$ samples an independently random vector $\mathbf{e}'_{\theta,\ell} \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell} = [\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}^*}]\mathbf{e}_{\theta,\ell}$ for $\ell \in [d]$.
- For $\theta \notin \mathsf{Path}(\mathtt{BT}, \eta^*)$, the creation of $\mathbf{u}'_{\theta,\ell}$ is the same as that of **Game** 1 for $\ell \in [d] \setminus \mathcal{F}_{\mathsf{T}^*}$. For the nodes, $\mathcal{C}$ samples $\tilde{\mathbf{e}}_\theta \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell}$ for $\ell \in \mathcal{F}_{\mathsf{T}^*}$ as uniformly random vectors in $\mathbb{Z}_q^n$ subject to $\mathbf{u} - \sum_{\ell\in\mathcal{F}_{\mathsf{T}^*}} \mathbf{u}'_{\theta,\ell} = [\mathbf{A}_0|\mathbf{F}_{\mathsf{T}^*}]\tilde{\mathbf{e}}_\theta$.

When $\mathcal{A}$ is type-II:

- For $\theta \in \mathsf{Path}(\mathtt{BT}, \eta^*)$, $\mathcal{C}$ samples an independently random vector $\mathbf{e}'_{\theta,\ell} \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell} = [\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}^*}]\mathbf{e}_{\theta,\ell}$ for $\ell \in [d] \setminus \{\ell^*\}$. For the nodes, $\mathcal{C}$ samples $\tilde{\mathbf{e}}_\theta \leftarrow \mathsf{SampleGaussian}(\mathbb{Z}^{2m}, \sigma)$ and sets $\mathbf{u}'_{\theta,\ell} = \mathbf{u} - \sum_{\ell\in\mathcal{F}_{\mathsf{T}^*}\setminus\{\ell^*\}} \mathbf{u}'_{\theta,\ell} - [\mathbf{A}_0|\mathbf{F}_{\mathsf{T}^*}]\tilde{\mathbf{e}}_\theta$.
- For $\theta \notin \mathsf{Path}(\mathtt{BT}, \eta^*)$, the creations of $\mathbf{u}'_{\theta,\ell}$ and $\tilde{\mathbf{e}}_\theta$ are the same as in **Game** 2 against the Type-I adversary.

For every $\ell \in [d]$, $\mathcal{B}$ samples $\mathbf{e}'_{\theta,\ell} \leftarrow \mathsf{SampleRight}(\mathbf{A}_0, H(\mathsf{ID})\mathbf{G}, \mathbf{R}_{\mathsf{ID}^*}, \mathbf{u}'_{\theta,\ell}, \mathbf{T}_{\mathbf{G}}, \sigma)$ for $\mathsf{ID} \neq \mathsf{ID}^*$ as stated in **Game** 3. For ID^*, $\mathcal{B}$ uses $\mathbf{e}'_{\theta,\ell}$ which it computes during the creations of $\mathbf{u}'_{\theta,\ell}$ as a part of $\mathsf{SK}_{\mathsf{ID}^*}$ as stated in **Game** 2. Then, $\mathcal{B}$ outputs $\mathsf{SK}_{\mathsf{ID}} = \left(\left\{\theta, \{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]}\right\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}\right)$ along with an updated binary tree $\mathtt{BT}'$ (and returns nothing to $\mathcal{A}$). When $\mathcal{A}$ is type-II, $\mathsf{SK}_{\mathsf{ID}^*} = \left(\left\{\theta, \{\mathbf{e}'_{\theta,\ell}\}_{\ell\in[d]\setminus\{\ell^*\}}\right\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}\right)$

Secret Key Reveal Query: Upon a query $\mathsf{ID} \in \mathcal{I}$ from $\mathcal{A}$, $\mathcal{B}$ checks if the conditions (a) and (b) are satisfied (see Section 3 for details). If these conditions are *not* satisfied, then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ finds $\mathsf{SK}_{\mathsf{ID}}$ from $\mathtt{SKList}$, and returns it to $\mathcal{A}$.

Revocation & Key Update Query: Upon a query $\mathsf{RL} \subseteq \mathcal{I}$ (which denotes the set of identities that are going to be revoked in the next time period) from $\mathcal{A}$, $\mathcal{B}$ checks if the conditions (a)–(c) are satisfied (see Section 3 for details). If these conditions are *not* satisfied, then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.
Otherwise $\mathcal{B}$ increments the current time period by $\mathsf{T_{cu}} \leftarrow \mathsf{T_{cu}} + 1$. Then, $\mathcal{B}$ updates $\mathsf{RL}_{\mathsf{T_{cu}}} \leftarrow \mathsf{RL}$ and runs $\mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T_{cu}}}) \leftarrow \mathsf{CS.Cover}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T_{cu}}})$. For each node $\theta \in \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T_{cu}}})$, recall $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \in (\mathbb{Z}_q^n)^d$ if they were defined. Otherwise, choose random $\{\mathbf{u}'_{\theta,\ell}\}_{\ell\in[d]} \xleftarrow{\$} (\mathbb{Z}_q^n)^d$ and store them in θ as during secret key generation queries. $\mathcal{B}$ samples $\tilde{\mathbf{e}}_\theta \leftarrow \mathsf{SampleRight}(\mathbf{A}_0, H(\mathsf{T_{cu}})\mathbf{G}, \mathbf{R}_{\mathsf{T}^*}, \mathbf{u} - \sum_{\ell\in\mathcal{F}_{\mathsf{T_{cu}}}} \mathbf{u}'_{\theta,\ell}, \mathbf{T}_{\mathbf{G}}, \sigma)$ for $\mathsf{T_{cu}} \neq \mathsf{T}^*$ as stated in **Game** 3. For T^*, $\mathcal{B}$ uses $\tilde{\mathbf{e}}_\theta$ which it computes during the creations of $\mathbf{u}'_{\theta,\ell}$ as a part of $\mathsf{KU}_{\mathsf{T}^*}$ as stated in **Game** 3. Then, $\mathcal{B}$ returns $\mathsf{KU}_{\mathsf{T_{cu}}} = \left(\{\theta, \tilde{\mathbf{e}}_\theta\}_{\theta\in\mathsf{KUNode}(\mathtt{BT},\mathsf{RL}_{\mathsf{T_{cu}}})}, \mathcal{F}_{\mathsf{T_{cu}}}\right)$ to $\mathcal{A}$.

Decryption Key Reveal Query: Upon a query $(\mathsf{ID}, \mathsf{T}) \in \mathcal{I} \times \mathcal{T}$ from $\mathcal{A}$, $\mathcal{B}$ checks if the conditions (a)–(d) are satisfied (see Section 3 for details). If these conditions are *not* satisfied,

then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ finds $\mathsf{SK}_{\mathsf{ID}}$ from $\mathtt{SKList}$, runs $\mathsf{DK}_{\mathsf{ID},\mathsf{T}} \leftarrow \mathsf{DKG}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{T}})$, and returns $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ to $\mathcal{A}$.

Challenge Query: Upon a query $M^* \in \mathcal{M}$ from $\mathcal{A}$, $\mathcal{B}$ computes

$$c_0 = v_0 + M^* \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q, \qquad \mathbf{c} = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{ID}^*}^\top \\ \mathbf{R}_{\mathsf{T}^*}^\top \end{bmatrix} \mathbf{v}' \in \mathbb{Z}_q^{3m}.$$

Finally, $\mathcal{B}$ returns the challenge ciphertext $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$ to $\mathcal{A}$.

Guess: At some point, $\mathcal{A}$ outputs $b' \in \{0,1\}$ as its guess for b and terminates. Then, $\mathcal{B}$ outputs $\mathcal{A}$'s guess as the answer to the LWE challenge. This completes the description of $\mathcal{B}$.

We show that the challenge ciphertext $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$ is properly distributed. At first we show that if the LWE instance follows the form $(\mathbf{u}|\mathbf{A}_0)^\top \mathbf{s} + \mathbf{x}$, then $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$ is distributed as in **Game** 3. Observe that

$$[\mathbf{A}_0|\mathbf{F}_{\mathsf{ID}^*}|\mathbf{F}_{\mathsf{T}^*}] = [\mathbf{A}_0|\mathbf{A}_0\mathbf{R}_{\mathsf{ID}^*}|\mathbf{A}_0\mathbf{R}_{\mathsf{T}^*}].$$

By definition, $v_0 = \mathbf{u}^\top \mathbf{s} + x$ with a noise $x \in \mathbb{Z}_q$ distributed as $\mathcal{D}_{\mathbb{Z},\alpha q}$ and $\mathbf{v}' = \mathbf{A}_0^\top \mathbf{s} + \mathbf{y}$ with a noise vector $\mathbf{y} \in \mathbb{Z}_q^m$ distributed as $\mathcal{D}_{\mathbb{Z}^m,\alpha q}$. Hence,

$$\begin{aligned} c_0 &= v_0 + M^* \left\lfloor \frac{q}{2} \right\rfloor = \mathbf{u}^\top \mathbf{s} + x + M^* \left\lfloor \frac{q}{2} \right\rfloor, \\ \mathbf{c} &= \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{ID}^*}^\top \\ \mathbf{R}_{\mathsf{T}^*}^\top \end{bmatrix} \mathbf{v}' = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{ID}^*}^\top \\ \mathbf{R}_{\mathsf{T}^*}^\top \end{bmatrix} (\mathbf{A}_0^\top \mathbf{s} + \mathbf{y}) = \begin{bmatrix} \mathbf{A}_0^\top \\ (\mathbf{A}_0\mathbf{R}_{\mathsf{ID}^*})^\top \\ (\mathbf{A}_0\mathbf{R}_{\mathsf{T}^*})^\top \end{bmatrix} \mathbf{s} + \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{ID}^*}^\top \\ \mathbf{R}_{\mathsf{T}^*}^\top \end{bmatrix} \mathbf{y} \\ &= [\mathbf{A}_0|\mathbf{A}_0\mathbf{R}_{\mathsf{ID}^*}|\mathbf{A}_0\mathbf{R}_{\mathsf{T}^*}]^\top \mathbf{s} + \begin{bmatrix} \mathbf{I}_m \\ \mathbf{R}_{\mathsf{ID}^*}^\top \\ \mathbf{R}_{\mathsf{T}^*}^\top \end{bmatrix} \mathbf{y}, \end{aligned}$$

holds, where the creation is the same as in **Game** 3.

Next, if the LWE instance $\mathbf{v}$ is a random vector, then $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$ is distributed as in $\mathbf{Game}^{final}$. Since v_0 is uniform in $\mathbb{Z}_q$, $c_0 = v_0 + M^* \left\lfloor \frac{q}{2} \right\rfloor$ is also uniform in $\mathbb{Z}_q$. Since $\mathbf{v}'$ is uniform in $\mathbb{Z}_q^m$, $\mathbf{c} = [\mathbf{I}_m|\mathbf{R}_{\mathsf{ID}^*}|\mathbf{R}_{\mathsf{T}^*}]^\top \mathbf{v}'$ is also uniform in $\mathbb{Z}_q^{3m}$ by the standard leftover hash lemma, e.g., Theorem 8.38 of [Sho06].

Thus, we complete the proof. □

4.3 Discussion

Towards Full DKER. After the publication of the preliminary version of this paper [TW17], Katsumata et al. [KMT18] constructed the first lattice-based RIBE scheme with full DKER, where the construction completely differs from ours. An advantage of our B-DKER scheme is that our scheme satisfies anonymity while Katsumata et al.'s scheme does not have.

Insecurity of Cheng-Zhang's RIBE Scheme [CZ15]. Cheng and Zhang claimed that their proposed RIBE scheme with the subset difference (SD) method is the first adaptively secure one

with smaller key updates. However, there are critical bugs in their security proof, i.e., Game 3 in the proof of their Theorem 1. Here, we follow the notation from [CZ15], e.g., id and t. In their Game 3, the simulator aborts the game if $h_{\mathsf{id}^*} = 0$, where $h_{()}$ is a certain function for achieving adaptive security, to answer secret key extraction queries. In addition, the simulator also aborts the game if $h_{\mathsf{id}^*} \neq 0$ to create a challenge ciphertext. Hence, the game never ends. Note that the same situation occurs for the target time period t^*.

One may think that Chen et al.'s Gaussian sampling technique [CLL+12b], which we also used, can be used to fix the bugs. However, it is not the case. Furthermore, Cheng-Zhang's RIBE scheme is not secure even in the selective security model. The difficulty comes from the SD method which they used to revoke users. The SD method is another subset cover framework and it enables us to reduce the size of key updates. Notice that the subset cover framework which Chen et al. [CLL+12b] and we used in this paper is the CS method. If we modify Cheng-Zhang's RIBE scheme in the selective security model, the secret key $\mathbf{e}'$ and the key update $\tilde{\mathbf{e}}$ satisfy the following equations:

$$[\mathbf{A}_0|\mathbf{A}_1 + H(\mathsf{id})\mathbf{G}]\,\mathbf{e}' = \mathbf{u}' \qquad \text{and} \qquad [\mathbf{A}_0|\mathbf{A}_2 + H(\mathsf{t})\mathbf{G}]\,\tilde{\mathbf{e}} = \tilde{\mathbf{u}}.$$

The main difference between the SD method and the CS method is the restriction of syndrome vectors $\mathbf{u}'$ and $\tilde{\mathbf{u}}$. In the security proof, the simulator should create both the secret key $\mathbf{e}'$ for the target id^* and the key update $\tilde{\mathbf{e}}$ for the target t^*. As opposed to the CS method case, if we use the SD method, the simulator should create both $\mathbf{e}'$ and $\tilde{\mathbf{e}}$ for the same syndrome vector $\mathbf{u}' = \tilde{\mathbf{u}}$ even without DKER. Since we cannot create the keys by using the trapdoor $\mathbf{T}_{\mathbf{G}}$, we try to create them by using a Gaussian sampling algorithm. Once the simulator uses a Gaussian sampling algorithm to sample $\mathbf{e}'$ for the target id^*, then the corresponding syndrome vector $\mathbf{u}' = \tilde{\mathbf{u}}$ is fixed. Therefore, the simulator cannot create $\tilde{\mathbf{e}}$ for the target t^* by using a Gaussian sampling algorithm. Therefore, a construction of lattice-based RIBE with the SD method even in the selective security model and even without DKER is an interesting open problem.

Semi-adaptive Security. If we replace the hash function $\mathbf{F}_{\mathsf{ID}} = \mathbf{A}_1 + H(\mathsf{ID})\mathbf{G}$ of Agrawal et al. [ABB10a] by that of adaptively secure schemes [AFL16, Boy10, BL16, CHKP12, GPV08, KY16, Yam16, Yam17, ZCZ16], our scheme achieves semi-adaptive security[5], where an adversary issues the target $(\mathsf{ID}^*, \mathsf{T}^*)$ in advance of any key queries. What is required to prove the security of lattice-based RIBE is an existence of trapdoors that can sample short vectors $\mathbf{e}'_{\theta,\ell}$ for $\mathsf{ID} \neq \mathsf{ID}^*$ and $\tilde{\mathbf{e}}$ for $\mathsf{T} \neq \mathsf{T}^*$ according to discrete Gaussian distributions, where all the lattice-based IBE schemes have. However, it is insufficient to construct adaptively secure RIBE even without DKER.

In the RIBE setting, we have to set all $\mathbf{u}'_{\theta,\ell}$ in advance of any key queries, then we use $\mathbf{F}_{\mathsf{ID}^*}$, or equivalently ID^*, for the computations. It means that the simulator has to know ID^* at that time. To avoid the obstacle, we should develop new lattice-based RIBE constructions, which are different from Chen et al.'s [CLL+12b], or it may be equivalent to new lattice-based fuzzy IBE constructions, which are different from Agrawal et al.'s [ABV+12].

[5] Notice that we do not have to replace $\mathbf{F}_{\mathsf{T}} = \mathbf{A}_2 + H(\mathsf{T})\mathbf{G}$ by adaptively secure ones. Since the maximum time period is polynomially bounded, $|\mathcal{T}|$ security loss enables us to guess the target time period T^*. Indeed, Seo-Emura [SE14b] constructed adaptively secure DKER RIBE scheme by combining the Waters IBE [Wat05] for ID and the Boneh-Boyen IBE [BB11] for T.

One may think that adaptively secure IBE is more than enough to construct semi-adaptively secure RIBE. However, we do not know how to construct semi-adaptively secure lattice-based IBE that is more efficient than adaptively secure ones. We think that the construction should be an interesting open problem in this research topic.

5 B-DKER RIBE Scheme from Pairings

5.1 Construction

In this section, we construct a pairing-based B-DKER RIBE scheme which is based on the Jutla-Roy IBE [JR17] and its variant [RS14]. Specifically, we show our construction based on the 2-level Jutla-Roy HIBE. Roughly speaking, we combine it with CFFs and the CS method, and use the second level of the Jutla-Roy HIBE for time-periods. As we will explain in Section 5.2.2, it is not straightforward to utilize the dual system encryption methodology [Wat09] for RIBE constructions. To overcome the issue, we carefully design the sequence of games, and utilize the property of CFFs.

Our RIBE construction is as follows:

$\mathsf{SetUp}(1^\lambda)$: Let $N := \mathsf{superpoly}(\lambda)$, where $\mathsf{superpoly}(\cdot)$ is some super-polynomial (e.g., $\lambda^{\log \lambda}$). Run $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow \mathcal{G}$, $(w, d, \mathcal{F}) \leftarrow \mathsf{CFF.Gen}(N, Q)$, and $\mathtt{BT} \leftarrow \mathsf{CS.SetUp}(N)$. Let $\mathsf{RL}_0 := \emptyset$. Choose $\alpha, x, x_0, x_1, x_2, x_3, y, y_0, y_1, y_2, y_3 \xleftarrow{\$} \mathbb{Z}_q$, and set

$$z := e(g_1, g_2)^{x_0\alpha + y_0}, \quad u_{\mathrm{ID}} := g_1^{x_1\alpha + y_1}, \quad u_{\mathrm{T}} := g_1^{x_2\alpha + y_2}, \quad h := g_1^{x_3\alpha + y_3}, \quad v := g_1^{x\alpha + y}.$$

Output

$$\begin{aligned}\mathsf{PP} &:= (\mathbb{G}, g_1^\alpha, u_{\mathrm{ID}}, u_{\mathrm{T}}, h, v, z)\,,\\ \mathsf{MK} &:= (\mathtt{BT}, \mathsf{RL}_0, x, x_0, x_1, x_2, x_3, y, y_0, y_1, y_2, y_3)\,,\end{aligned}$$

$\mathsf{PKG}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$: Parse MK as $(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}, x, x_0, x_1, x_2, x_3, y, y_0, y_1, y_2, y_3)$. Run $(\eta_{\mathsf{ID}}, \mathtt{BT}') \leftarrow \mathsf{CS.Assign}(\mathtt{BT}, \mathsf{ID})$. For each node $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})$, recall $\{P_{\theta,x}^{(\ell)}, P_{\theta,y}^{(\ell)}\}_{\ell=1}^d$ if they were defined. Otherwise, choose random $\{P_{\theta,x}^{(\ell)}, P_{\theta,y}^{(\ell)}\}_{\ell=1}^d \in G_2$ and store them in θ. For every $\ell \in [d]$, choose $r_{\theta,\ell} \in \mathbb{Z}_q$ and compute

$$\begin{aligned}\mathtt{sk}_{\mathsf{ID},\theta}^{(\ell)} &:= g_2^{r_{\theta,\ell}},\\ \mathtt{sk}_{\mathsf{ID},\theta,x}^{(\ell)} := g_2^{r_{\theta,\ell}x}, \quad \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,x} &:= P_{\theta,x}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)}, \quad \mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,x} := g_2^{r_{\theta,\ell}x_2},\\ \mathtt{sk}_{\mathsf{ID},\theta,y}^{(\ell)} := g_2^{r_{\theta,\ell}y}, \quad \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,y} &:= P_{\theta,y}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)}, \quad \mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,y} := g_2^{r_{\theta,\ell}y_2}.\end{aligned}$$

Set $\mathsf{SK}_{\mathsf{ID},\theta}^{(\ell)} := \left(\mathtt{sk}_{\mathsf{ID},\theta}^{(\ell)}, \mathtt{sk}_{\mathsf{ID},\theta,x}^{(\ell)}, \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,x}, \mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,x}, \mathtt{sk}_{\mathsf{ID},\theta,y}^{(\ell)}, \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,y}, \mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,y}\right)$. Output

$$\begin{aligned}\mathsf{SK}_{\mathsf{ID}} &= \left\{\mathsf{SK}_{\mathsf{ID},\theta} := \left(\theta, \left\{\mathsf{SK}_{\mathsf{ID},\theta}^{(\ell)}\right\}_{\ell\in[d]}\right)\right\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})},\\ \mathsf{MK}' &:= (\mathtt{BT}', \mathsf{RL}_{\mathsf{T}}, x, x_0, x_1, x_2, x_3, y, y_0, y_1, y_2, y_3).\end{aligned}$$

$\mathsf{KeyUp}(\mathsf{PP}, \mathsf{T}, \mathsf{MK})$: Parse MK as $(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}, x, x_0, x_1, x_2, x_3, y, y_0, y_1, y_2, y_3)$. Run $\mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}) \leftarrow \mathsf{CS.Cover}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$. For each node $\theta \in \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$, recall $\{P_{\theta,x}^{(\ell)}, P_{\theta,y}^{(\ell)}\}_{\ell=1}^{d}$ if they were defined. Otherwise, choose random $\{P_{\theta,x}^{(\ell)}, P_{\theta,y}^{(\ell)}\}_{\ell=1}^{d} \in G_2$ and store them in θ. Choose $s_\theta \in \mathbb{Z}_q$ and compute

$$\begin{aligned}
\mathtt{ku}_{\mathsf{T},\theta} &:= g_2^{s_\theta}, \\
\mathtt{ku}_{\mathsf{T},\theta,x} &:= g_2^{s_\theta x}, \quad \mathtt{ku}'_{\mathsf{T},\theta,x} := \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}}} \left(P_{\theta,x}^{(\ell)}\right)^{-1}\right) \cdot g_2^{x_0 + s_\theta(x_2 \mathsf{T} + x_3)}, \quad \mathtt{ku}''_{\mathsf{T},\theta,x} := g_2^{s_\theta x_1}, \\
\mathtt{ku}_{\mathsf{T},\theta,y} &:= g_2^{s_\theta y}, \quad \mathtt{ku}'_{\mathsf{T},\theta,y} := \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}}} \left(P_{\theta,y}^{(\ell)}\right)^{-1}\right) \cdot g_2^{y_0 + s_\theta(y_2 \mathsf{T} + y_3)}, \quad \mathtt{ku}''_{\mathsf{T},\theta,y} := g_2^{s_\theta y_1}.
\end{aligned}$$

Set $\mathsf{KU}_{\mathsf{T},\theta} := \left(\theta, \mathtt{ku}_{\mathsf{T},\theta}, \mathtt{ku}_{\mathsf{T},\theta,x}, \mathtt{ku}'_{\mathsf{T},\theta,x}, \mathtt{ku}''_{\mathsf{T},\theta,x}, \mathtt{ku}_{\mathsf{T},\theta,y}, \mathtt{ku}'_{\mathsf{T},\theta,y}, \mathtt{ku}''_{\mathsf{T},\theta,y}\right)$. Output

$$\begin{aligned}
\mathsf{KU}_{\mathsf{T}} &:= \left(\{\mathsf{KU}_{\mathsf{T},\theta}\}_{\theta \in \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})}, \mathcal{F}_{\mathsf{T}}\right), \\
\mathsf{MK}' &:= (\mathtt{BT}', \mathsf{RL}_{\mathsf{T}}, x, x_0, x_1, x_2, x_3, y, y_0, y_1, y_2, y_3).
\end{aligned}$$

$\mathsf{DKG}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{KU}_{\mathsf{T}})$: Parse $\mathsf{SK}_{\mathsf{ID}}$, and KU_{T} as $\{(\theta, \{\mathsf{SK}_{\mathsf{ID},\theta}^{(\ell)}\}_{\ell \in [d]})\}_{\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})}$ and $(\{\mathsf{KU}_{\mathsf{T},\theta}\}_{\theta \in \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})}, \mathcal{F}_{\mathsf{T}})$, respectively. Output $\perp$ if $\emptyset \leftarrow \mathsf{CS.Match}(\mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}}), \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}))$. Otherwise, for $\theta \leftarrow \mathsf{CS.Match}(\mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}}), \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}}))$, compute

$$\begin{aligned}
\mathtt{dk}_{\mathsf{ID},\mathsf{T}} &:= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}}} \mathtt{sk}_{\mathsf{ID},\theta}^{(\ell)}\right) \cdot \mathsf{KU}_{\mathsf{T},\theta} = g_2^{\sum_{\ell \in \mathcal{F}_{\mathsf{T}}} r_{\theta,\ell} + s_\theta} = g_2^{r}, \\
\mathtt{dk}_{\mathsf{ID},\mathsf{T},x} &= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}}} \mathtt{sk}_{\mathsf{ID},\theta,x}^{(\ell)}\right) \cdot \mathsf{KU}_{\mathsf{T},\theta,x} = g_2^{(\sum_{\ell \in \mathcal{F}_{\mathsf{T}}} r_{\theta,\ell} + s_\theta)x} = g_2^{rx}, \\
\mathtt{dk}'_{\mathsf{ID},\mathsf{T},x} &= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}}} \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,x} \left(\mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,x}\right)^{\mathsf{T}}\right) \cdot \mathsf{KU}'_{\mathsf{T},\theta,x} \left(\mathsf{KU}''_{\mathsf{T},\theta,x}\right)^{\mathsf{ID}} \\
&= g_2^{x_0 + (\sum_{\ell \in \mathcal{F}_{\mathsf{T}}} r_{\theta,\ell} + s_\theta)(x_1 \mathsf{ID} + x_2 \mathsf{T} + x_3)} = g_2^{x_0 + r(x_1 \mathsf{ID} + x_2 \mathsf{T} + x_3)}, \\
\mathtt{dk}_{\mathsf{ID},\mathsf{T},y} &= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}}} \mathtt{sk}_{\mathsf{ID},\theta,y}^{(\ell)}\right) \cdot \mathsf{KU}_{\mathsf{T},\theta,y} = g_2^{(\sum_{\ell \in \mathcal{F}_{\mathsf{T}}} r_{\theta,\ell} + s_\theta)y} = g_2^{ry}, \\
\mathtt{dk}'_{\mathsf{ID},\mathsf{T},y} &= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}}} \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,y} \left(\mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,y}\right)^{\mathsf{T}}\right) \cdot \mathsf{KU}'_{\mathsf{T},\theta,y} \left(\mathsf{KU}''_{\mathsf{T},\theta,y}\right)^{\mathsf{ID}} \\
&= g_2^{y_0 + (\sum_{\ell \in \mathcal{F}_{\mathsf{T}}} r_{\theta,\ell} + s_\theta)(y_1 \mathsf{ID} + y_2 \mathsf{T} + y_3)} = g_2^{y_0 + r(y_1 \mathsf{ID} + y_2 \mathsf{T} + y_3)},
\end{aligned}$$

where $\mathsf{SK}^{(\ell)}_{\mathsf{ID},\theta} = (\mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta}, \mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta,x}, \mathtt{sk}'^{(\ell)}_{\mathsf{T},\theta,x}, \mathtt{sk}''^{(\ell)}_{\mathsf{T},\theta,x}, \mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta,y}, \mathtt{sk}'^{(\ell)}_{\mathsf{T},\theta,y}, \mathtt{sk}''^{(\ell)}_{\mathsf{T},\theta,y})$, $\mathsf{KU}_{\mathsf{T},\theta} = (\theta, \mathtt{ku}_{\mathsf{T},\theta}, \mathtt{ku}_{\mathsf{T},\theta,x}, \mathtt{ku}'_{\mathsf{T},\theta,x}, \mathtt{ku}''_{\mathsf{T},\theta,x}, \mathtt{ku}_{\mathsf{T},\theta,y}, \mathtt{ku}'_{\mathsf{T},\theta,y}, \mathtt{ku}''_{\mathsf{T},\theta,y})$, and $r := \sum_{\ell \in \mathcal{F}_{\mathsf{T}}} r_{\theta,\ell} + s_\theta$. Output

$$\mathsf{DK}_{\mathsf{ID},\mathsf{T}} := (\mathtt{dk}_{\mathsf{ID},\mathsf{T}}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},y}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},y}).$$

$\mathsf{Enc}(\mathsf{PP}, \mathsf{ID}, \mathsf{T}, M \in G_T)$: Pick random $t, \delta_1, \delta_2, \delta_3 \in \mathbb{Z}_q$ and compute

$$\mathtt{tag} := \delta_1 \mathsf{ID} + \delta_2 \mathsf{T} + \delta_3, \quad \mathtt{ct}_{\mathrm{M}} := M \cdot z^t, \quad \mathtt{ct}_x := (g_1^\alpha)^t, \quad \mathtt{ct}_y := g_1^t, \quad \mathtt{ct}_{\mathsf{ID},\mathsf{T}} := \left(v^{\mathtt{tag}} u_{\mathrm{ID}}^{\mathsf{ID}} u_{\mathrm{T}}^{\mathsf{T}} h\right)^t.$$

Output $\mathsf{CT}_{\mathsf{ID},\mathsf{T}} := (\mathtt{ct}_{\mathrm{M}}, \mathtt{ct}_x, \mathtt{ct}_y, \mathtt{ct}_{\mathsf{ID},\mathsf{T}}, \mathtt{tag})$.

$\mathsf{Dec}(\mathsf{PP}, \mathsf{DK}_{\mathsf{ID},\mathsf{T}}, \mathsf{CT}_{\mathsf{ID},\mathsf{T}})$: Parse $\mathsf{DK}_{\mathsf{ID},\mathsf{T}}$ and $\mathsf{CT}_{\mathsf{ID},\mathsf{T}}$ as $(\mathtt{dk}_{\mathsf{ID},\mathsf{T}}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},y}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},y})$ and $(\mathtt{ct}_{\mathrm{M}}, \mathtt{ct}_x, \mathtt{ct}_y, \mathtt{ct}_{\mathsf{ID},\mathsf{T}}, \mathtt{tag})$, respectively. Compute

$$M = \frac{\mathtt{ct}_{\mathrm{M}} \cdot e(\mathtt{ct}_{\mathsf{ID},\mathsf{T}}, \mathtt{dk}_{\mathsf{ID},\mathsf{T}})}{e(\mathtt{ct}_x, \mathtt{dk}^{\mathtt{tag}}_{\mathsf{ID},\mathsf{T},x} \mathtt{dk}'_{\mathsf{ID},\mathsf{T},x}) \cdot e(\mathtt{ct}_y, \mathtt{dk}^{\mathtt{tag}}_{\mathsf{ID},\mathsf{T},y} \mathtt{dk}'_{\mathsf{ID},\mathsf{T},y})}.$$

Correctness. We show the decryption correctness. For $\mathsf{DK}_{\mathsf{ID},\mathsf{T}} = (\mathtt{dk}_{\mathsf{ID},\mathsf{T}}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},y}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},y})$ and $\mathsf{CT}_{\mathsf{ID},\mathsf{T}} = (\mathtt{ct}_{\mathrm{M}}, \mathtt{ct}_x, \mathtt{ct}_y, \mathtt{ct}_{\mathsf{ID},\mathsf{T}}, \mathtt{tag})$, we have

$$\begin{aligned}
&\frac{\mathtt{ct}_{\mathrm{M}} \cdot e(\mathtt{ct}_{\mathsf{ID},\mathsf{T}}, \mathtt{dk}_{\mathsf{ID},\mathsf{T}})}{e(\mathtt{ct}_x, \mathtt{dk}^{\mathtt{tag}}_{\mathsf{ID},\mathsf{T},x} \mathtt{dk}'_{\mathsf{ID},\mathsf{T},x}) \cdot e(\mathtt{ct}_y, \mathtt{dk}^{\mathtt{tag}}_{\mathsf{ID},\mathsf{T},y} \mathtt{dk}'_{\mathsf{ID},\mathsf{T},y})} \\
&= \frac{\mathtt{ct}_{\mathrm{M}} \cdot e\left(\left((g_1^{x\alpha+y})^{\mathtt{tag}} (g_1^{x_1\alpha+y_1})^{\mathsf{ID}} (g_1^{x_2\alpha+y_2})^{\mathsf{T}} g_1^{x_3\alpha+y_3}\right)^t, g_2^r\right)}{e((g_1^\alpha)^t, (g_2^{rx})^{\mathtt{tag}} g_2^{x_0 + r(x_1\mathsf{ID} + x_2\mathsf{T} + x_3)}) \cdot e(g_1^t, (g_2^{ry})^{\mathtt{tag}} g_2^{y_0 + r(y_1\mathsf{ID} + y_2\mathsf{T} + y_3)})} \\
&= \frac{\mathtt{ct}_{\mathrm{M}} \cdot e\left(\left(g_1^{\mathtt{tag}(x\alpha+y) + \mathsf{ID}(x_1\alpha+y_1) + \mathsf{T}(x_2\alpha+y_2) + x_3\alpha+y_3)}\right)^t, g_2^r\right)}{e((g_1^\alpha)^t, g_2^{x_0 + r(x\mathtt{tag} + x_1\mathsf{ID} + x_2\mathsf{T} + x_3)}) \cdot e(g_1^t, g_2^{y_0 + r(y\mathtt{tag} + y_1\mathsf{ID} + y_2\mathsf{T} + y_3)})} \\
&= \frac{M \cdot e(g_1, g_2)^{t(x_0\alpha + y_0)} \cdot e(g_1^{\mathtt{tag}(x\alpha+y) + \mathsf{ID}(x_1\alpha+y_1) + \mathsf{T}(x_2\alpha+y_2) + x_3\alpha+y_3}, g_2)^{rt}}{e(g_1^\alpha, g_2^{x_0})^t \cdot e(g_1^\alpha, g_2^{x\mathtt{tag} + x_1\mathsf{ID} + x_2\mathsf{T} + x_3})^{rt} \cdot e(g_1, g_2^{y_0})^t \cdot e(g_1, g_2^{y\mathtt{tag} + y_1\mathsf{ID} + y_2\mathsf{T} + y_3})^{rt}} \\
&= M.
\end{aligned}$$

5.2 Security

The proposed construction can be proved to be IND-RID-Q-CPA secure.

Theorem 2. *If the SXDH assumption holds and the underlying CFF is Q-cover-free, then the proposed RIBE scheme is IND-RID-Q-CPA secure.*

Note that our pairing-based construction does not require the underlying CFF to be w-uniform. We devote the rest of this section to the proof of the above theorem.

5.2.1 Semi-functional Ciphertexts, Keys, and Key Updates

We use the dual system encryption methodology to prove Theorem 2. We describe semi-functional forms of ciphertexts, secret keys, key updates, and decryption keys.

Semi-functional ciphertext for $(\mathsf{ID}, \mathsf{T})$: For a normal ciphertext $\mathsf{CT}_{\mathsf{ID},\mathsf{T}} = (\mathtt{ct}_{\mathrm{M}}, \mathtt{ct}_x, \mathtt{ct}_y, \mathtt{ct}_{\mathsf{ID},\mathsf{T}}, \mathtt{tag})$, its semi-functional form $\widetilde{\mathsf{CT}}_{\mathsf{ID},\mathsf{T}} = (\widetilde{\mathtt{ct}}_{\mathrm{M}}, \widetilde{\mathtt{ct}}_x, \widetilde{\mathtt{ct}}_y, \widetilde{\mathtt{ct}}_{\mathsf{ID},\mathsf{T}}, \widetilde{\mathtt{tag}})$ is computed by

$$\begin{aligned}
\widetilde{\mathtt{tag}} &:= \mathtt{tag},\\
\widetilde{\mathtt{ct}}_{\mathrm{M}} &:= \mathtt{ct}_{\mathrm{M}} \cdot e(g_1, g_2)^{x_0 \mu} = M \cdot e(g_1, g_2)^{x_0(t\alpha+\mu)+y_0 t},\\
\widetilde{\mathtt{ct}}_x &:= \mathtt{ct}_x \cdot g_1^{\mu} = g_1^{t\alpha+\mu},\\
\widetilde{\mathtt{ct}}_y &:= \mathtt{ct}_y = g_1^{t},\\
\widetilde{\mathtt{ct}}_{\mathsf{ID},\mathsf{T}} &:= \mathtt{ct}_{\mathsf{ID},\mathsf{T}} \cdot g_1^{\mu(x\widetilde{\mathtt{tag}}+x_1\mathsf{ID}+x_2\mathsf{T}+x_3)} = g_1^{(t\alpha+\mu)(x\widetilde{\mathtt{tag}}+x_1\mathsf{ID}+x_2\mathsf{T}+x_3)+t(y\widetilde{\mathtt{tag}}+y_1\mathsf{ID}+y_2\mathsf{T}+y_3)},
\end{aligned}$$

where $\mu \xleftarrow{\$} \mathbb{Z}_q$.

Semi-functional secret-key for ID: Let η be a leaf node of $\mathtt{BT}$ assigned to ID, and parse a normal secret key $\mathsf{SK}_{\mathsf{ID}}$ as $\{\mathsf{SK}_{\mathsf{ID},\theta} = (\theta, \{\mathsf{SK}_{\mathsf{ID},\theta}^{(\ell)}\}_{\ell\in[d]})\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}$. For each $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})$ and $\ell \in [d]$, a semi-functional secret-key component $\widetilde{\mathsf{SK}}_{\mathsf{ID},\theta}^{(\ell)} := (\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta}^{(\ell)}, \widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,x}^{(\ell)}, \widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,x}^{(\ell)}, \widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,x}^{(\ell)}, \widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,y}^{(\ell)}, \widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,y}^{(\ell)}, \widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,y}^{(\ell)})$ is computed by

$$\begin{aligned}
\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta}^{(\ell)} &:= \mathtt{sk}_{\mathsf{ID},\theta}^{(\ell)},\\
\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,x}^{(\ell)} &:= \mathtt{sk}_{\mathsf{ID},\theta,x}^{(\ell)},\\
\widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,x}^{(\ell)} &:= \mathtt{sk}'_{\mathsf{ID},\theta,x}^{(\ell)} \cdot g_2^{\phi_{\theta,\ell}} = P_{\theta,x}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)+\phi_{\theta,\ell}},\\
\widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,x}^{(\ell)} &:= \mathtt{sk}''_{\mathsf{ID},\theta,x}^{(\ell)},\\
\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,y}^{(\ell)} &:= \mathtt{sk}_{\mathsf{ID},\theta,y}^{(\ell)},\\
\widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,y}^{(\ell)} &:= \mathtt{sk}'_{\mathsf{ID},\theta,y}^{(\ell)} \cdot g_2^{-\alpha\phi_{\theta,\ell}} = P_{\theta,y}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)-\alpha\phi_{\theta,\ell}},\\
\widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,y}^{(\ell)} &:= \mathtt{sk}''_{\mathsf{ID},\theta,y}^{(\ell)},
\end{aligned}$$

where $\mathsf{SK}_{\mathsf{ID},\theta}^{(\ell)} = (\mathtt{sk}_{\mathsf{ID},\theta}^{(\ell)}, \mathtt{sk}_{\mathsf{ID},\theta,x}^{(\ell)}, \mathtt{sk}'_{\mathsf{ID},\theta,x}^{(\ell)}, \mathtt{sk}''_{\mathsf{ID},\theta,x}^{(\ell)}, \mathtt{sk}_{\mathsf{ID},\theta,y}^{(\ell)}, \mathtt{sk}'_{\mathsf{T},\theta,y}^{(\ell)}, \mathtt{sk}''_{\mathsf{T},\theta,y}^{(\ell)})$ and $\phi_{\theta,\ell} \xleftarrow{\$} \mathbb{Z}_q$. A semi-functional secret key is $\widetilde{\mathsf{SK}}_{\mathsf{ID}} := \{\widetilde{\mathsf{SK}}_{\mathsf{ID},\theta} = (\theta, \{\widetilde{\mathsf{SK}}_{\mathsf{ID},\theta}^{(\ell)}\}_{\ell\in[d]})\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}$.

Semi-functional key update for T: Parse a normal key update KU_{T} as $(\{\mathsf{KU}_{\mathsf{T},\theta}\}_{\theta\in\mathsf{KUNode}(\mathtt{BT},\mathsf{RL}_{\mathsf{T}})}, \mathcal{F}_{\mathsf{T}})$. For each $\theta \in \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}})$, a semi-functional key-update component $\widetilde{\mathsf{KU}}_{\mathsf{T},\theta} := (\theta, \widetilde{\mathtt{ku}}_{\mathsf{T},\theta}, \widetilde{\mathtt{ku}}_{\mathsf{T},\theta,x}, \widetilde{\mathtt{ku}'}_{\mathsf{T},\theta,x}, \widetilde{\mathtt{ku}''}_{\mathsf{T},\theta,x}, \widetilde{\mathtt{ku}}_{\mathsf{T},\theta,y}, \widetilde{\mathtt{ku}'}_{\mathsf{T},\theta,y}, \widetilde{\mathtt{ku}''}_{\mathsf{T},\theta,y})$ is computed by

$$\widetilde{\mathtt{ku}}_{\mathsf{T},\theta} := \mathtt{ku}_{\mathsf{T},\theta},$$

$$
\begin{aligned}
\widetilde{\mathtt{ku}}_{\mathsf{T},\theta,x} &:= \mathtt{ku}_{\mathsf{T},\theta,x},\\
\widetilde{\mathtt{ku}'}_{\mathsf{T},\theta,x} &:= \mathtt{ku}'_{\mathsf{T},\theta,x} \cdot g_2^{\psi_\theta} = \left(\prod_{\ell\in\mathcal{F}} \left(P_{\theta,x}^{(\ell)}\right)^{-1}\right) \cdot g_2^{x_0+s_\theta(x_2\mathsf{T}+x_3)+\psi_\theta},\\
\widetilde{\mathtt{ku}''}_{\mathsf{T},\theta,x} &:= \mathtt{ku}''_{\mathsf{T},\theta,x},\\
\widetilde{\mathtt{ku}}_{\mathsf{T},\theta,y} &:= \mathtt{ku}_{\mathsf{T},\theta,y},\\
\widetilde{\mathtt{ku}'}_{\mathsf{T},\theta,y} &:= \mathtt{ku}'_{\mathsf{T},\theta,y} \cdot g_2^{-\alpha\psi_\theta} = \left(\prod_{\ell\in\mathcal{F}} \left(P_{\theta,y}^{(\ell)}\right)^{-1}\right) \cdot g_2^{y_0+s_\theta(y_2\mathsf{T}+y_3)-\alpha\psi_\theta},\\
\widetilde{\mathtt{ku}''}_{\mathsf{T},\theta,y} &:= \mathtt{ku}''_{\mathsf{T},\theta,y},
\end{aligned}
$$

where $\mathsf{KU}_{\mathsf{T},\theta} = (\theta, \mathtt{ku}_{\mathsf{T},\theta}, \mathtt{ku}_{\mathsf{T},\theta,x}, \mathtt{ku}'_{\mathsf{T},\theta,x}, \mathtt{ku}''_{\mathsf{T},\theta,x}, \mathtt{ku}_{\mathsf{T},\theta,y}, \mathtt{ku}'_{\mathsf{T},\theta,y}, \mathtt{ku}''_{\mathsf{T},\theta,y})$ and $\psi_\theta \xleftarrow{\$} \mathbb{Z}_q$. A semi-functional key update is $\widetilde{\mathsf{KU}}_{\mathsf{T}} := (\{\widetilde{\mathsf{KU}}_{\mathsf{T},\theta}\}_{\theta\in\mathsf{KUNode}(\mathsf{BT},\mathsf{RL}_{\mathsf{T}})}, \mathcal{F}_{\mathsf{T}})$.

Semi-functional decryption key for (ID,T): A decryption key for (ID,T) is semi-functional if a secret key for ID and/or a key update for T input into the DKG algorithm is semi-functional. More specifically, a semi-functional decryption key $\widetilde{\mathsf{DK}}_{\mathsf{ID},\mathsf{T}} := (\widetilde{\mathtt{dk}}_{\mathsf{ID},\mathsf{T}}, \widetilde{\mathtt{dk}}_{\mathsf{ID},\mathsf{T},x}, \widetilde{\mathtt{dk}'}_{\mathsf{ID},\mathsf{T},x}, \widetilde{\mathtt{dk}}_{\mathsf{ID},\mathsf{T},y}, \widetilde{\mathtt{dk}'}_{\mathsf{ID},\mathsf{T},y})$ is

$$
\begin{aligned}
\widetilde{\mathtt{dk}}_{\mathsf{ID},\mathsf{T}} &:= \mathtt{dk}_{\mathsf{ID},\mathsf{T}},\\
\widetilde{\mathtt{dk}}_{\mathsf{ID},\mathsf{T},x} &:= \mathtt{dk}_{\mathsf{ID},\mathsf{T},x}, \quad \widetilde{\mathtt{dk}'}_{\mathsf{ID},\mathsf{T},x} := \mathtt{dk}'_{\mathsf{ID},\mathsf{T},x} \cdot g_2^{\varphi} = g_2^{x_0+r(x_1\mathsf{ID}+x_2\mathsf{T}+x_3)+\varphi},\\
\widetilde{\mathtt{dk}}_{\mathsf{ID},\mathsf{T},y} &:= \mathtt{dk}_{\mathsf{ID},\mathsf{T},y}, \quad \widetilde{\mathtt{dk}'}_{\mathsf{ID},\mathsf{T},y} := \mathtt{dk}'_{\mathsf{ID},\mathsf{T},y} \cdot g_2^{-\alpha\varphi} = g_2^{y_0+r(y_1\mathsf{ID}+y_2\mathsf{T}+y_3)-\alpha\varphi},
\end{aligned}
$$

where $\mathsf{DK}_{\mathsf{ID},\mathsf{T}} = (\mathtt{dk}_{\mathsf{ID},\mathsf{T}}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},x}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},y}, \mathtt{dk}'_{\mathsf{ID},\mathsf{T},y})$ is a normal decryption key and $\varphi \in \mathbb{Z}_q$.

Note that $g_1^{x_0}$ and g_2^{α} are needed to compute semi-functional ciphertexts and keys (including key updates), respectively. The following equation ensures that a normal ciphertext for (ID,T) can be decrypted by a semi-functional decryption key for (ID,T):

$$e(\mathtt{ct}_x, g_2^{\varphi}) \cdot e(\mathtt{ct}_y, g_2^{-\alpha\varphi}) = 1_{G_T},$$

where 1_{G_T} is an identity element in G_T. The following equation also ensures that a semi-functional ciphertext for (ID,T) can be decrypted by a normal decryption key for (ID,T):

$$\frac{e(g_1,g_2)^{x_0\mu} \cdot e(g_1^{\mu(x\widetilde{\mathtt{tag}}+x_1\mathsf{ID}+x_2\mathsf{T}+x_3)})}{e(g_1^{\mu}, \mathtt{dk}_{\mathsf{ID},\mathsf{T},x}^{\widetilde{\mathtt{tag}}}\mathtt{dk}'_{\mathsf{ID},\mathsf{T},x})} = 1_{G_T}.$$

5.2.2 Proof Idea and Game Sequence

Our proof approach. As Lee [Lee16] pointed out, we have to carefully design the hybrid games when using the dual system encryption methodology in the RIBE (with DKER) setting. In security proofs of all the existing (H)IBE schemes employing dual system encryption, secret keys queried to the key extraction oracle are changed into semi-functional ones one by one. Similarly, we need to

Table 1: The overview of the hybrid games.

	Challenger knows T^* and i^*?	$\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$	$\mathsf{SK}_{\mathsf{ID}}$ for $\mathsf{ID} \neq \mathsf{ID}^*$	KU_{T} for $\mathsf{T} \neq \mathsf{T}^*$	$\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$
Game^{real}	no	normal	normal	normal	normal
Game 0	yes	normal	normal	normal	normal
Game 1	yes	sf	normal	normal	normal
Game 2	yes	sf	sf	normal	normal
Game 3	yes	sf	sf	sf	normal
Game 4	yes	sf	sf	sf	sf
Game^{final}	yes	random	sf	sf	sf

transform both secret keys and key updates into semi-functional forms. However, in RIBE, unlike (H)IBE, a secret key for the target identity ID^* and/or a key update for the target time period T^* may be issued to the oracles, and it might cause a bug in the security proof.

Our approach is similar to Lee's one (also see Table 1): First, we assume we know when the target time period T^* is and when the target identity ID^* is queried, and suppose that an i^*-th secret key reveal query for $PKG(\cdot)$ is ID^* (**Game** 0). It enables us to apply the strategy-dividing lemma (Lemma 7). As in the dual-system-encryption (H)IBE schemes, we can change secret keys for $\mathsf{ID} \neq \mathsf{ID}^*$ and key updates for $\mathsf{T} \neq \mathsf{T}^*$ into their semi-functional forms without any problem (**Game** 2 and **Game** 3). Then, we show that the distributions on a normal secret key $\mathsf{SK}_{\mathsf{ID}^*}$ for ID^* and a normal key update $\mathsf{KU}_{\mathsf{T}^*}$ for T^* are identical to those on semi-functional ones ($\widetilde{\mathsf{SK}}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$). Namely, we show **Game** 3 and **Game** 4 are identical. As the final transition, we replace the challenge ciphertext with the random elements of the ciphetext space, i.e., $G_T \times G_1^3 \times \mathbb{Z}_q$ (**Game**^{final}). It ensures that the adversary's advantage is exactly $1/2$.

Formal description of the game sequence. Formally, the hybrid games are defined as follows.

Game^{real}**:** The original IND-RID-Q-CPA game.

Game 0**:** The game is the same as **Game**^{real} except that at the beginning of the game, the challenger correctly guesses the challenge time period T^* and i^* such that $\mathsf{ID}_{i^*} = \mathsf{ID}^*$.

Game 1**:** The game is the same as **Game** 0 except that the challenge ciphertext $\mathsf{CT}_{\mathsf{ID}^*,\mathsf{T}^*}$ is semi-functional.

Game 2_j $(1 \leq j \leq Q_{sk})$**:** Let $\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_{Q_{sk}}$. **Game** 2_j is the same as **Game** 2_{j-1} except that for a query ID_j, the semi-functional form of $\mathsf{SK}_{\mathsf{ID}_j}$ is returned if $\mathsf{ID}_j \neq \mathsf{ID}^*$, where **Game** 2_0

means **Game** 1.

Game 3_j $(1 \leq j \leq |\mathcal{T}|)$**:** **Game** 3_j is the same as **Game** 3_{j-1} except that for a j-th revocation and key update query, the semi-functional form of KU_j is returned if $j \neq \mathsf{T}^*$, where **Game** 3_0 means **Game** $2_{Q_{sk}}$.

Game 4**:** This game is the same as **Game** $3_{|\mathcal{T}|}$ except that for a T^*-th revocation and key update query (resp., a secret key reveal query ID^*), the semi-functional form of $\mathsf{KU}_{\mathsf{T}^*}$ (resp., $\mathsf{SK}_{\mathsf{ID}^*}$) is returned.

Game^{final}**:** This game is the same as **Game** 4 except that the challenge ciphertext is random element in the ciphertext space.

Let E_k be an event that $b' = b$ occurs in **Game** k or **Game**^{k}. We then have

$$\begin{aligned}
&\frac{1}{2}\mathsf{Adv}^{\mathit{IND\text{-}Q\text{-}CPA}}_{\Pi,\mathcal{A}}(\lambda) \\
&\quad = \left|\Pr[E_{real}] - \frac{1}{2}\right| \\
&\quad \leq |\Pr[E_{real}] - \Pr[E_0]| + |\Pr[E_0] - \Pr[E_1]| + \sum_{j=0}^{Q_{sk}-1} \left|\Pr[E_{2_j}] - \Pr[E_{2_{j+1}}]\right| \\
&\qquad\qquad + \sum_{j=0}^{|\mathcal{T}|-1} \left|\Pr[E_{3_j}] - \Pr[E_{3_{j+1}}]\right| + \left|\Pr[E_{3_{|\mathcal{T}|}}] - \Pr[E_4]\right| + |\Pr[E_4] - \Pr[E_{final}]| \,.
\end{aligned}$$

Note that $|\Pr[E_{final}] - 1/2| = 0$.

5.2.3 Proof of Theorem 2

Without loss of generality, in the proof we assume an adversary $\mathcal{A}$ issues only $(\mathsf{ID}^*, \cdot)$ as decryption key reveal queries since $\mathcal{A}$ can get any decryption keys for ID $(\neq \mathsf{ID}^*)$ from $\mathsf{SK}_{\mathsf{ID}}$ and KU_{T} obtained from the challenger $\mathcal{C}$. We also assume that the running time of an adversary $\mathcal{A}$ is publicly known (or, can be easily estimated), and the maximum number of queries issued to all the oracles by $\mathcal{A}$ (denoted by Q_{all}) can be estimated from $\mathcal{A}$'s running time.[6]

The difference between **Game** 0 and **Game**^{real} is whether or not the challenger knows (T^*, i^*) at the beginning of the game. First, we consider the probability that the challenger $\mathcal{C}$ correctly guesses i^* such that $\mathsf{ID}_{i^*} = \mathsf{ID}^*$. Let I_C be a random variable of the challenger's guess, which takes values in $\{0, 1, \ldots, Q_{all}\}$. "$I_C = 0$" indicates that $\mathcal{C}$ guesses that $\mathcal{A}$ never issues ID^* to the *PKG* oracle (i.e., $\mathcal{A}$ is a Type-II adversary). Similarly, let $I_{\mathcal{A}}$ be a random variable of $\mathcal{A}$'s choice, which tales values in $\{0, 1, \ldots, Q_{all}\}$. We then have

$$\Pr[I_C = I_{\mathcal{A}}]$$

[6] In practice, we may estimate Q_{all} based on the running time of the fastest (super)computer, and such Q_{all} is still a polynomial in λ.

$$
\begin{aligned}
&= \Pr[I_C = 0 \wedge I_{\mathcal{A}} = 0] + \cdots + \Pr[I_C = Q_{all} \wedge I_{\mathcal{A}} = Q_{all}] \\
&= \frac{1}{Q_{all}+1} \Pr[I_{\mathcal{A}} = 0 \mid I_C = 0] + \cdots + \frac{1}{Q_{all}+1} \Pr[I_{\mathcal{A}} = Q_{all} \mid I_C = Q_{all}] \\
&= \frac{1}{Q_{all}+1} \sum_{j=0}^{Q_{all}} \Pr[I_{\mathcal{A}} = j] \qquad (4) \\
&= \frac{1}{Q_{all}+1},
\end{aligned}
$$

where Eq. (4) follows from that $\mathcal{C}$ guesses i^* independently of $\mathcal{A}'$'s choice. Similarly, we can prove $\mathcal{C}$ correctly guesses T^* with probability $1/|\mathcal{T}|$.

Therefore, the challenger's guess is right with probability $1/(|\mathcal{T}|(Q_{all}+1))$. In other words, the reduction loss is $|\mathcal{T}|(Q_{all}+1)$, which is polynomial in the security parameter. In the rest of the proof, the simulator knows the exact value of T^*, and can distinguish if each queried identity is the challenge one or not.

The difference between **Game** 0 and **Game** 1 is whether the challenge ciphertext is normal or semi-functional. The following lemma can be proved in a similar way to the Jutla-Roy (H)IBE [JR17, RS14].

Lemma 8. *If there exists a PPT adversary $\mathcal{A}$ to distinguish* **Game** 1 *and* **Game** 0*, then there exists a PPT adversary $\mathcal{B}$ to break the DDH*1 *assumption.*

Proof. Given the DDH1 assumption instance $(\mathbb{G}, g_1^{c_1}, g_1^{c_2})$ with Z. The simulator $\mathcal{B}$ uses $\mathcal{A}$ to distinguish if Z is distributed as $g_1^{c_1 c_2}$ or $g_1^{c_1 c_2 + \mu}$.

$\mathcal{B}$ implicitly sets $\alpha := c_1$, and chooses $x, x_0, x_1, x_2, x_3, y, y_0, y_1, y_2, y_3 \xleftarrow{\$} \mathbb{Z}_q$. $\mathcal{B}$ then computes

$$
\begin{aligned}
&v := (g_1^{c_1})^x g_1^y, \quad z := e(g_1^{c_1}, g_2)^{x_0} e(g_1, g_2)^{y_0}, \\
&u_{\mathrm{ID}} := (g_1^{c_1})^{x_1} g_1^{y_1}, \quad u_{\mathrm{T}} := (g_1^{c_1})^{x_2} g_1^{y_2}, \quad h := (g_1^{c_1})^{x_3} g_1^{y_3}.
\end{aligned}
$$

$\mathcal{B}$ runs $\mathsf{CS.SetUp}(N) \to \mathtt{BT}$ by choosing arbitrary $N \in \mathbb{N}$, and sends $\mathsf{PP} := (\mathbb{G}, g_1^{\alpha}, u_{\mathrm{ID}}, u_{\mathrm{T}}, h, v, z)$ to $\mathcal{A}$. Note that $\mathcal{B}$ knows the master key. Therefore, $\mathcal{B}$ can respond to any queries.

When $\mathcal{A}$ submits $(M^*, \mathsf{ID}^*, \mathsf{T}^*)$, $\mathcal{B}$ picks $\mathtt{tag}^* \xleftarrow{\$} \mathbb{Z}_q$, and creates the challenge ciphertext $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{T}^*} := (\mathtt{ct}^*_{\mathrm{M}}, \mathtt{ct}^*_x, \mathtt{ct}^*_y, \mathtt{ct}^*_{\mathsf{ID}^*,\mathsf{T}^*}, \mathtt{tag}^*)$ as

$$
\begin{aligned}
&\mathtt{ct}^*_{\mathrm{M}} := M^* \cdot e(Z, g_2)^{x_0} e(g_1^{c_2}, g_2)^{y_0}, \quad \mathtt{ct}^*_x := Z, \quad \mathtt{ct}^*_y := g_1^{c_2}, \\
&\mathtt{ct}^*_{\mathsf{ID}^*,\mathsf{T}^*} := Z^{x\mathtt{tag}^* + x_1 \mathsf{ID}^* + x_2 \mathsf{T}^* + x_3} (g_1^{c_2})^{y\mathtt{tag}^* + y_1 \mathsf{ID}^* + y_2 \mathsf{T}^* + y_3}.
\end{aligned}
$$

Obviously, the above ciphertext is normal if $Z = g_1^{c_1 c_2}$. We show the above ciphertext is semi-functional if $Z = g_1^{c_1 c_2 + \mu}$ as follows.

$$
\begin{aligned}
\mathtt{ct}^*_{\mathrm{M}} &= M^* \cdot e(Z, g_2)^{x_0} e(g_1^{c_2}, g_2)^{y_0} \\
&= M^* \cdot e(g_1^{c_1 c_2 + \mu})^{x_0} e(g_1^{c_2}, g_2)^{y_0} \\
&= M^* \cdot e(g_1)^{c_2(x_0 c_1 + y_0)} e(g_1, g_2)^{x_0 \mu}
\end{aligned}
$$

$$
\begin{aligned}
&= M^* \cdot z^t \cdot e(g_1, g_2)^{x_0\mu}, \\
\mathtt{ct}^*_x &= Z = g_1^{c_1c_2+\mu} = g_1^{\alpha t+\mu}, \\
\mathtt{ct}^*_y &= g_1^{c_2} = g_1^t, \\
\mathtt{ct}^*_{\mathsf{ID},\mathsf{T}} &= Z^{x\mathtt{tag}^*+x_1\mathsf{ID}^*+x_2\mathsf{T}^*+x_3} g_1^{y\mathtt{tag}^*+y_1\mathsf{ID}^*+y_2\mathsf{T}^*+y_3} \\
&= g_1^{(c_1c_2+\mu)(x\mathtt{tag}^*+x_1\mathsf{ID}^*+x_2\mathsf{T}^*+x_3)} g_1^{c_2(y\mathtt{tag}^*+y_1\mathsf{ID}^*+y_2\mathsf{T}^*+y_3)} \\
&= g_1^{c_2)((xc_1+y_1)\mathtt{tag}^*+(x_1c_1+y_1)\mathsf{ID}^*+(x_2c_1+y_2)\mathsf{T}^*+x_3c_1+y_3)} g_1^{\mu(x\mathtt{tag}^*+x_1\mathsf{ID}^*+x_2\mathsf{T}^*+x_3)} \\
&= \left(v^{\mathtt{tag}^*} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t g_1^{\mu(x\mathtt{tag}^*+x_1\mathsf{ID}^*+x_2\mathsf{T}^*+x_3)}.
\end{aligned}
$$

□

In a similar way to ordinary the dual-system-encryption IBE schemes, we can show how normal secret keys for any ID ($\neq \mathsf{ID}^*$) and key updates for any T ($\neq \mathsf{T}^*$) are transformed into their semi-functional forms. Therefore, we omit the proofs here, and they are given in Appendix A.

We next show the distributions in **Game** $3_{|\mathcal{T}|}$ and **Game** 4 are identical.

Lemma 9. **Game** $3_{|\mathcal{T}|}$ *and* **Game** 4 *are identical from the viewpoint of any PPT adversary* $\mathcal{A}$.

Proof. The difference between **Game** $3_{|\mathcal{T}|}$ and **Game** 4 is whether a secret key for ID^* and a key update for T^* are normal or semi-functional. Therefore, the corresponding decryption keys for $(\mathsf{ID}^*, \mathsf{T})$ also become normal or semi-functional depending on **Game** $3_{|\mathcal{T}|}$ or **Game** 4.

Consider the following components of $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$:

$$
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}^*,\theta,x} := \widetilde{P}^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}^*+x_3)+\beta_{\theta,\ell}}, \tag{5}
$$

$$
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}^*,\theta,y} := \widetilde{P}^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}^*+y_3)-\alpha\beta_{\theta,\ell}}, \tag{6}
$$

$$
\mathtt{ku}'_{\mathsf{T}^*,\theta,x} := \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}^*}} \left(\widetilde{P}^{(\ell)}_{\theta,x}\right)^{-1}\right) \cdot g_2^{x_0+s_\theta(x_2\mathsf{T}^*+x_3)-\sum_{\ell\in\mathcal{F}_{\mathsf{T}^*}}\beta'_{\theta,\ell}}, \tag{7}
$$

$$
\mathtt{ku}'_{\mathsf{T}^*,\theta,y} := \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}^*}} \left(\widetilde{P}^{(\ell)}_{\theta,y}\right)^{-1}\right) \cdot g_2^{y_0+s_\theta(y_2\mathsf{T}^*+y_3)-\alpha\left(-\sum_{\ell\in\mathcal{F}_{\mathsf{T}^*}}\beta'_{\theta,\ell}\right)}. \tag{8}
$$

Note that in both games, all other secret keys and key updates are semi-functional. Therefore, other secret key and key update components corresponding to the above θ (i.e., $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}^*}) \cup \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}^*})$) are

$$
\mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,x} := \widetilde{P}^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)+\phi_{\theta,\ell}}, \tag{9}
$$

$$
\mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,y} := \widetilde{P}^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)-\alpha\phi_{\theta,\ell}}, \tag{10}
$$

$$
\mathtt{ku}'_{\mathsf{T},\theta,x} := \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}} \left(\widetilde{P}^{(\ell)}_{\theta,x}\right)^{-1}\right) \cdot g_2^{x_0+s_\theta(x_2\mathsf{T}^*+x_3)+\psi_\theta}, \tag{11}
$$

$$\mathtt{ku}'_{\mathsf{T},\theta,y} := \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}}\left(\widetilde{P}^{(\ell)}_{\theta,y}\right)^{-1}\right)\cdot g_2^{y_0+s_\theta(y_2\mathsf{T}^*+y_3)-\alpha\psi_\theta}. \tag{12}$$

We show the distributions on the above (Eqs. (5)–(12)) are actually identical to the distributions in both **Game** $3_{|\mathcal{T}|}$ and **Game** 4 from the viewpoint of an adversary $\mathcal{A}$. Specifically, we show the above from the standpoint of each of $\mathcal{A}_I$ and $\mathcal{A}_{II}$.

Type-I adversary $\mathcal{A}_I$: For every $\theta \in \mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}^*}) \cup \mathsf{KUNode}(\mathtt{BT},\mathsf{RL}_{\mathsf{T}^*})$ and $\ell \in [d]$, if we set

$$P^{(\ell)}_{\theta,x} = \widetilde{P}^{(\ell)}_{\theta,x}\cdot g_2^{\beta_{\theta,\ell}} \text{ and } P^{(\ell)}_{\theta,y} = \widetilde{P}^{(\ell)}_{\theta,y}\cdot g_2^{-\alpha\beta_{\theta,\ell}},$$

it is easy to see that Eqs. (5)–(8) then turn to normal components of $\mathsf{SK}_{\mathsf{ID}^*}$ and $\mathsf{KU}_{\mathsf{T}^*}$. Then, we show that the distributions on other secret keys and key updates are also identical to those in **Game** $3_{|\mathcal{T}|}$. Since $\mathsf{CS.Match}(\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}^*}),\mathsf{KUNode}(\mathtt{BT},\mathsf{RL}_{\mathsf{T}^*})) = \emptyset$, all the components of $\widetilde{\mathsf{SK}}_{\mathsf{ID}}$ ($\mathsf{ID}\neq\mathsf{ID}^*$) and $\widetilde{\mathsf{KU}}_{\mathsf{T}}$ ($\mathsf{T}\neq\mathsf{T}^*$) corresponding to $\theta \in \mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}^*}) \cup \mathsf{KUNode}(\mathtt{BT},\mathsf{RL}_{\mathsf{T}^*})$ are semi-functional. Namely, Eqs. (9)–(12) turn to

$$\begin{aligned}
\mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,x} &= \widetilde{P}^{(\ell)}_{\theta,x}\cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)+\phi_{\theta,\ell}}\\
&= P^{(\ell)}_{\theta,x}\cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)+\phi_{\theta,\ell}-\beta_{\theta,\ell}} = P^{(\ell)}_{\theta,x}\cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)+\tilde{\phi}_{\theta,\ell}},\\
\mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,y} &:= \widetilde{P}^{(\ell)}_{\theta,y}\cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)-\alpha\phi_{\theta,\ell}}\\
&= P^{(\ell)}_{\theta,y}\cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)-\alpha(\phi_{\theta,\ell}-\beta_{\theta,\ell})} = P^{(\ell)}_{\theta,y}\cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)-\alpha\tilde{\phi}_{\theta,\ell}},\\
\mathtt{ku}'_{\mathsf{T},\theta,x} &= \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}}\left(\widetilde{P}^{(\ell)}_{\theta,x}\right)^{-1}\right)\cdot g_2^{x_0+s_\theta(x_2\mathsf{T}^*+x_3)+\psi_\theta}\\
&= \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}}\left(P^{(\ell)}_{\theta,x}\right)^{-1}\right)\cdot g_2^{x_0+s_\theta(x_2\mathsf{T}^*+x_3)+\psi_\theta+\sum_{\ell\in\mathcal{F}_{\mathsf{T}}}\beta_{\theta,\ell}}\\
&= \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}}\left(P^{(\ell)}_{\theta,x}\right)^{-1}\right)\cdot g_2^{x_0+s_\theta(x_2\mathsf{T}^*+x_3)+\tilde{\psi}_\theta},\\
\mathtt{ku}'_{\mathsf{T},\theta,y} &= \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}}\left(\widetilde{P}^{(\ell)}_{\theta,y}\right)^{-1}\right)\cdot g_2^{y_0+s_\theta(y_2\mathsf{T}^*+y_3)-\alpha\psi_\theta}\\
&= \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}}\left(P^{(\ell)}_{\theta,y}\right)^{-1}\right)\cdot g_2^{x_0+s_\theta(x_2\mathsf{T}^*+x_3)-\alpha\left(\psi_\theta+\sum_{\ell\in\mathcal{F}_{\mathsf{T}}}\beta_{\theta,\ell}\right)}\\
&= \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}}}\left(P^{(\ell)}_{\theta,y}\right)^{-1}\right)\cdot g_2^{x_0+s_\theta(x_2\mathsf{T}^*+x_3)-\alpha\tilde{\psi}_\theta},
\end{aligned}$$

where $\tilde{\phi}_{\theta,\ell} := \phi_{\theta,\ell} - \beta_{\theta,\ell}$ and $\tilde{\psi}_\theta := \psi_\theta + \sum_{\ell \in \mathcal{F}_{\mathsf{T}}} \beta_{\theta,\ell}$. Since $\phi_{\theta,\ell}$ and ψ_θ are randomly chosen for every node θ of $\mathtt{BT}$ and $\ell \in [d]$, the distributions on the above components are identical to those in **Game** $3_{|\mathcal{T}|}$.

On the other hand, for every $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}^*}) \cup \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}^*})$, if we set

$$P^{(\ell)}_{\theta,x} = \widetilde{P}^{(\ell)}_{\theta,x} \text{ and } P^{(\ell)}_{\theta,y} = \widetilde{P}^{(\ell)}_{\theta,y},$$

Eqs. (5)–(8) then turn to semi-functional components of $\widetilde{\mathsf{SK}}_{\mathsf{ID}^*}$ and $\widetilde{\mathsf{KU}}_{\mathsf{T}^*}$ since we have

$$\begin{aligned}
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}^*,\theta,x} &= \widetilde{P}^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1 \mathsf{ID}^* + x_3) + \beta_{\theta,\ell}} = P^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1 \mathsf{ID}^* + x_3) + \phi_{\theta,\ell}}, \\
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}^*,\theta,y} &= \widetilde{P}^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1 \mathsf{ID}^* + y_3) - \alpha\beta_{\theta,\ell}} = P^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1 \mathsf{ID}^* + y_3) - \alpha\phi_{\theta,\ell}}, \\
\mathtt{ku}'_{\mathsf{T}^*,\theta,x} &= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \left(\widetilde{P}^{(\ell)}_{\theta,x} \right)^{-1} \right) \cdot g_2^{x_0 + s_\theta(x_2 \mathsf{T}^* + x_3) - \sum_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \beta'_{\theta,\ell}} \\
&= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \left(P^{(\ell)}_{\theta,x} \right)^{-1} \right) \cdot g_2^{x_0 + s_\theta(x_2 \mathsf{T}^* + x_3) + \psi_\theta}, \\
\mathtt{ku}'_{\mathsf{T}^*,\theta,y} &= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \left(\widetilde{P}^{(\ell)}_{\theta,y} \right)^{-1} \right) \cdot g_2^{y_0 + s_\theta(y_2 \mathsf{T}^* + y_3) - \alpha\left(- \sum_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \beta'_{\theta,\ell} \right)} \\
&= \left(\prod_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \left(P^{(\ell)}_{\theta,y} \right)^{-1} \right) \cdot g_2^{y_0 + s_\theta(y_2 \mathsf{T}^* + y_3) - \alpha\psi_\theta},
\end{aligned}$$

where $\phi_{\theta,\ell} := \beta_{\theta,\ell}$ and $\psi_\theta := -\sum_{\ell \in \mathcal{F}_{\mathsf{T}^*}} \beta'_{\theta,\ell}$. Therefore, the distributions on Eqs. (5)–(12) are identical to those in **Game** 4.

Type-II adversary $\mathcal{A}_{II}$: Since $\mathcal{A}_{II}$ cannot get the secret key for ID^*, we only need to show the distributions on Eqs. (7)–(12) are identical to those in both **Game** $3_{|\mathcal{T}|}$ and **Game** 4.

In fact, we can show the distributions on Eqs. (7)–(12) are identical those in **Game** $3_{|\mathcal{T}|}$ as above. However, since $\mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}^*}) \cap \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}^*}) \neq \emptyset$, we have to be more careful about the distributions on decryption keys when we show that the distributions on Eqs. (7)–(12) are also identical to those in **Game** 4. Therefore, for every $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}^*}) \cup \mathsf{KUNode}(\mathtt{BT}, \mathsf{RL}_{\mathsf{T}^*})$, we set

$$\begin{cases}
P^{(\ell)}_{\theta,x} := \widetilde{P}^{(\ell)}_{\theta,x} \text{ and } P^{(\ell)}_{\theta,y} = \widetilde{P}^{(\ell)}_{\theta,y}, & \text{if } \ell \in \bigcup_{i=1}^{Q} \mathcal{F}_{\mathsf{T}_i}, \\
P^{(\ell)}_{\theta,x} := \widetilde{P}^{(\ell)}_{\theta,x} \cdot g_2^{\beta_{\theta,\ell}} \text{ and } P^{(\ell)}_{\theta,y} = \widetilde{P}^{(\ell)}_{\theta,y} \cdot g_2^{-\alpha\beta_{\theta,\ell}}, & \text{otherwise,}
\end{cases}$$

where $\mathsf{T}_1, \ldots, \mathsf{T}_Q$ are time periods issued for decryption key reveal queries. Then, there exists at least one element $\ell \in \mathcal{F}_{\mathsf{T}^*} \setminus \left(\bigcup_{i=1}^{Q} \mathcal{F}_{\mathsf{T}_i} \right)$ due to the underlying CFF, and therefore Eqs.

(7) and (8) turn to semi-functional by setting $\psi_\theta := -\sum_{\ell \in \mathcal{F}_{\mathsf{T}^*} \setminus (\bigcup_{i=1}^{Q} \mathcal{F}_{\mathsf{T}_i})} \beta_{\theta,\ell}$. Obviously, a decryption key $\widetilde{\mathsf{DK}}_{\mathsf{ID}^*,\mathsf{T}_i}$ for $(\mathsf{ID}^*, \mathsf{T}_i)$ with $i \in [Q]$ is semi-functional, since $\widetilde{\mathsf{KU}}_{\mathsf{T}_i}$ is semi-functional. As in the case of $\mathcal{A}_I$, we can also show that Eqs. (9)–(12) turn to semi-functional components, and the distributions on them are identical to those in **Game** 4. Hence, the distributions on Eqs. (7)–(12) are identical to those in **Game** 4.

This completes the proof. □

Finally, we show that the difference between **Game** 4 and **Game**^{final} is negligible as follows. First, we define the following sub-game **Game**$^{semi\text{-}final}$.

- **Game**$^{semi\text{-}final}$: **Game**$^{semi\text{-}final}$ is the same as **Game** 4 except that the first component $\widetilde{\mathsf{ct}}_{\mathrm{M}}$ of the semi-functional challenge ciphertext is the random element in $\mathbf{G}_T$.

We show that,

$$|\Pr[E_4] - \Pr[E_{final}]| \leq |\Pr[E_4] - \Pr[E_{semi\text{-}final}]| + |\Pr[E_{semi\text{-}final}] - \Pr[E_{final}]|,$$

is negligible.

Lemma 10. *If there exists a PPT adversary $\mathcal{A}$ to distinguish* **Game**$^{semi\text{-}final}$ *and* **Game** 4*, then there exists a PPT adversary $\mathcal{B}$ to break the DDH*1 *assumption.*

Proof. Given the DDH1 instance $(\mathbb{G}, g_1^{c_1}, g_1^{c_2})$ with Z. The simulator $\mathcal{B}$ uses $\mathcal{A}$ to distinguish if Z is distributed as $g_1^{c_1 c_2}$ or $g_1^{c_1 c_2 + \eta}$.

$\mathcal{B}$ chooses $\alpha, x, x_1, x_2, x_3, y, y_1, y_2, y_3 \xleftarrow{\$} \mathbb{Z}_q$. $\mathcal{B}$ also chooses $y_0' \xleftarrow{\$} \mathbb{Z}_q$ and implicitly sets

$$x_0 := c_1, \quad y_0 := y_0' - \alpha c_1.$$

$\mathcal{B}$ computes $z := e(g_1, g_2)^{y_0'}$, $v := g_1^{x\alpha+y}$, $u_{\mathrm{ID}} := g_1^{x_1\alpha+y_1}$, $u_{\mathrm{T}} := g_1^{x_2\alpha+y_2}$, and $h := g_1^{x_3\alpha+y_3}$. $\mathcal{B}$ runs $\mathsf{CS.SetUp}(N) \to \mathsf{BT}$ by arbitrarily choosing $N \in \mathbb{N}$, and sends $\mathsf{PP} := (\mathbb{G}, g_1^\alpha, u_{\mathrm{ID}}, u_{\mathrm{T}}, h, v, z)$ to $\mathcal{A}$.

$\mathcal{B}$ can respond to any secret key reveal queries since any secret key does not contain x_0 and y_0. Since correct semi-functional decryption keys for $(\mathsf{ID}^*, \mathsf{T})$ can be generated if $\mathcal{B}$ simulates correct semi-functional key updates, we here only show how $\mathcal{B}$ respond to revocation and key update queries as follows. When a receiving a query $\mathsf{RL} \subseteq \mathcal{I}$, $\mathcal{B}$ checks if the conditions (a)–(c) are satisfied (see Section 3 for details). If not, $\mathcal{B}$ outputs $\perp$. Otherwise, $\mathcal{B}$ sets $\mathsf{T}_{\mathsf{cu}} \leftarrow \mathsf{T}_{\mathsf{cu}} + 1$ and $\mathsf{RL}_{\mathsf{T}_{\mathsf{cu}}} \leftarrow \mathsf{RL}$, and then for every $\theta \in \mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}_{\mathsf{T}_{\mathsf{cu}}})$, $\mathcal{B}$ computes $\widetilde{\mathsf{KU}}_{\mathsf{T}_{\mathsf{cu}},\theta} := (\theta, \widetilde{\mathsf{ku}}_{\mathsf{T}_{\mathsf{cu}},\theta}, \widetilde{\mathsf{ku}}_{\mathsf{T}_{\mathsf{cu}},\theta,x}, \widetilde{\mathsf{ku}'}_{\mathsf{T}_{\mathsf{cu}},\theta,x}, \widetilde{\mathsf{ku}''}_{\mathsf{T}_{\mathsf{cu}},\theta,x}, \widetilde{\mathsf{ku}}_{\mathsf{T}_{\mathsf{cu}},\theta,y}, \widetilde{\mathsf{ku}'}_{\mathsf{T}_{\mathsf{cu}},\theta,y}, \widetilde{\mathsf{ku}''}_{\mathsf{T}_{\mathsf{cu}},\theta,y})$ as follows. $\mathcal{B}$ chooses $s_\theta, \psi_\theta' \xleftarrow{\$} \mathbb{Z}_q$, and implicitly sets

$$\psi_\theta := \psi_\theta' - x_0 \quad (\text{and hence } \psi_\theta' = x_0 + \psi_\theta).$$

$\mathcal{B}$ then computes

$\widetilde{\mathsf{ku}}_{\mathsf{T}_{\mathsf{cu}},\theta} := g_2^{s_\theta}$,

$$\widetilde{\mathtt{ku}}_{\mathsf{T}_{\mathrm{cu}},\theta,x} := g_2^{s_\theta x},$$
$$\widetilde{\mathtt{ku}'}_{\mathsf{T}_{\mathrm{cu}},\theta,x} := \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}_{\mathrm{cu}}}} \left(P_{\theta,x}^{(\ell)}\right)^{-1}\right) \cdot g_2^{s_\theta(x_2\mathsf{T}_{\mathrm{cu}}+x_3)+\psi'_\theta} = \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}_{\mathrm{cu}}}} \left(P_{\theta,x}^{(\ell)}\right)^{-1}\right) \cdot g_2^{x_0+s_\theta(x_2\mathsf{T}_{\mathrm{cu}}+x_3)+\psi_\theta},$$
$$\widetilde{\mathtt{ku}''}_{\mathsf{T}_{\mathrm{cu}},\theta,x} := g_2^{s_\theta x_1},$$
$$\widetilde{\mathtt{ku}}_{\mathsf{T}_{\mathrm{cu}},\theta,y} := g_2^{s_\theta y},$$
$$\widetilde{\mathtt{ku}'}_{\mathsf{T}_{\mathrm{cu}},\theta,y} := \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}_{\mathrm{cu}}}} \left(P_{\theta,y}^{(\ell)}\right)^{-1}\right) \cdot g_2^{y'_0+s_\theta(y_2\mathsf{T}_{\mathrm{cu}}+y_3)-\alpha\psi'_\theta} = \left(\prod_{\ell\in\mathcal{F}_{\mathsf{T}_{\mathrm{cu}}}} \left(P_{\theta,y}^{(\ell)}\right)^{-1}\right) \cdot g_2^{y_0+s_\theta(y_2\mathsf{T}_{\mathrm{cu}}+y_3)-\alpha\psi_\theta},$$
$$\widetilde{\mathtt{ku}''}_{\mathsf{T}_{\mathrm{cu}},\theta,y} := g_2^{s_\theta y_1}.$$

Since ψ'_θ masks x_0 in the information-theoretic sense, ψ_θ is a random element of $\mathbb{Z}_q$ from $\mathcal{A}$'s view. $\mathcal{B}$ returns $\widetilde{\mathsf{KU}}_{\mathsf{T}_{\mathrm{cu}}} := (\{\widetilde{\mathsf{KU}}_{\mathsf{T}_{\mathrm{cu}},\theta}\}_{\theta\in\mathsf{KUNode}(\mathsf{BT},\mathsf{RL}_{\mathsf{T}_{\mathrm{cu}}})}, \mathcal{F}_{\mathsf{T}_{\mathrm{cu}}})$ to $\mathcal{A}$.

When $\mathcal{A}$ submits $(M^*, \mathsf{ID}^*, \mathsf{T}^*)$, $\mathcal{B}$ picks $t, \delta_1^*, \delta_2^*, \delta_3^* \xleftarrow{\$} \mathbb{Z}_q$, and creates the challenge ciphertext $\widetilde{\mathsf{CT}}^*_{\mathsf{ID}^*,\mathsf{T}^*} := (\widetilde{\mathtt{ct}}^*_{\mathrm{M}}, \widetilde{\mathtt{ct}}^*_x, \widetilde{\mathtt{ct}}^*_y, \widetilde{\mathtt{ct}}^*_{\mathsf{ID}^*,\mathsf{T}^*}, \widetilde{\mathtt{tag}}^*)$ as

$$\begin{aligned}
&\widetilde{\mathtt{tag}}^* := \delta_1^*\mathsf{ID}^* + \delta_2^*\mathsf{T}^* + \delta_3^*,\\
&\widetilde{\mathtt{ct}}^*_{\mathrm{M}} := M^* \cdot z^t \cdot e(Z, g_2), \quad \widetilde{\mathtt{ct}}^*_x := (g_1^\alpha)^t g_1^{c_2}, \quad \widetilde{\mathtt{ct}}^*_y := g_1^t,\\
&\widetilde{\mathtt{ct}}^*_{\mathsf{ID},\mathsf{T}} := \left(v^{\widetilde{\mathtt{tag}}^*} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t (g_1^{c_2})^{x\widetilde{\mathtt{tag}}^* + x_1\mathsf{ID}^* + x_2\mathsf{T}^* + x_3}.
\end{aligned}$$

Obviously, by setting $\mu := c_2$ the above ciphertext is semi-functional if $Z = g_1^{c_1c_2}$. On the other hand, the first component of the above ciphertext is random element of $\mathbf{G}_1$ if $Z = g_1^{c_1c_2+\eta}$ since we have

$$\widetilde{\mathtt{ct}}^*_{\mathrm{M}} := M^* \cdot z^t \cdot e(Z, g_2) = M^* \cdot z^t \cdot e(g_1, g_2)^{x_0\mu+\eta} = R \cdot z^t \cdot e(g_1, g_2)^{x_0\mu},$$

where $R := M^* \cdot e(g_1, g_2)^\eta$. □

Lemma 11. *If there exists a PPT adversary $\mathcal{A}$ to distinguish* $\mathbf{Game}^{final}$ *and* $\mathbf{Game}^{semi\text{-}final}$*, then there exists a PPT adversary $\mathcal{B}$ to break the DDH1 assumption.*

Proof. Basically, this lemma can be proved in a similar way to Lemma 10.

The given DDH1 instance is $(\mathbb{G}, g_1^{c_1}, g_1^{c_2})$ with Z, where Z is $g_1^{c_1c_2}$ or $g_1^{c_1c_2+\eta}$. $\mathcal{B}$ chooses $\alpha, x, x_0, x_1, x_2, y, y_0, y_1, y_2 \xleftarrow{\$} \mathbb{Z}_q$. $\mathcal{B}$ also chooses $y'_3 \xleftarrow{\$} \mathbb{Z}_q$ and implicitly sets

$$x_3 := c_1, \quad y_3 := y'_3 - \alpha c_1.$$

$\mathcal{B}$ computes $z := e(g_1, g_2)^{x_0\alpha+y_0}$, $v := g_1^{x\alpha+y}$, $u_{\mathrm{ID}} := g_1^{x_1\alpha+y_1}$, $u_{\mathrm{T}} := g_1^{x_2\alpha+y_2}$, and $h := g_1^{y'_3}$. $\mathcal{B}$ runs $\mathsf{BT} \leftarrow \mathsf{CS.SetUp}(N)$ by choosing arbitrary $N \in \mathbb{N}$, and sends $\mathsf{PP} := (\mathbb{G}, g_1^\alpha, u_{\mathrm{ID}}, u_{\mathrm{T}}, h, v, z)$ to $\mathcal{A}$.

Here, we only show how $\mathcal{B}$ responds to secret key generation and secret key reveal queries since $\mathcal{B}$ can answer revocation and key update queries in a similar way. When receiving a query ID, $\mathcal{B}$ runs $(\eta_{\mathsf{ID}}, \mathtt{BT}) \leftarrow \mathsf{CS.Assign}(\mathtt{BT}, \mathsf{ID})$. For every $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})$, $\mathcal{B}$ recalls $\{P_{\theta,x}^{(\ell)}, P_{\theta,y}^{(\ell)}\}_{\ell=1}^{d}$ if they were defined. Otherwise, $\mathcal{B}$ randomly chooses them from G_2, and stores them in θ. For $\ell \in [d]$, $\mathcal{B}$ computes $\widetilde{\mathsf{SK}}_{\mathsf{ID},\theta}^{(\ell)} := (\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta}^{(\ell)}, \widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,x}^{(\ell)}, \widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,x}^{(\ell)}, \widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,x}^{(\ell)}, \widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,y}^{(\ell)}, \widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,y}^{(\ell)}, \widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,y}^{(\ell)})$ as follows. $\mathcal{B}$ chooses $r_{\theta,\ell}, \phi'_{\theta,\ell} \in \mathbb{Z}_q$, and implicitly sets

$$\phi_{\theta,\ell} := \phi'_{\theta,\ell} - r_{\theta,\ell} x_3 \ \ \text{(and hence } \phi'_{\theta,\ell} = r_{\theta,\ell} x_3 + \phi_{\theta,\ell}\text{)}.$$

Since $\phi'_{\theta,\ell}$ masks $r_{\theta,\ell} x_3$ in the information-theoretic sense, $\phi_{\theta,\ell}$ is a random element of $\mathbb{Z}_q$ from $\mathcal{A}$'s view. $\mathcal{B}$ then computes

$$\begin{aligned}
\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta}^{(\ell)} &:= g_2^{r_{\theta,\ell}},\\
\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,x}^{(\ell)} &:= g_2^{r_{\theta,\ell} x},\\
\widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,x}^{(\ell)} &:= P_{\theta,x}^{(\ell)} g_2^{r_{\theta,\ell} x_1 \mathsf{ID} + \phi'_{\theta,\ell}} = P_{\theta,x}^{(\ell)} g_2^{r_{\theta,\ell}(x_1 \mathsf{ID} + x_3) + \phi_{\theta,\ell}},\\
\widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,x}^{(\ell)} &:= g_2^{r_{\theta,\ell} x_2},\\
\widetilde{\mathtt{sk}}_{\mathsf{ID},\theta,y}^{(\ell)} &:= g_2^{r_{\theta,\ell} y},\\
\widetilde{\mathtt{sk}'}_{\mathsf{ID},\theta,y}^{(\ell)} &:= P_{\theta,y}^{(\ell)} g_2^{r_{\theta,\ell}(y_1 \mathsf{ID} + y'_3) - \alpha \phi'_{\theta,\ell}} = P_{\theta,y}^{(\ell)} g_2^{r_{\theta,\ell}(y_1 \mathsf{ID} + y_3) - \alpha \phi_{\theta,\ell}},\\
\widetilde{\mathtt{sk}''}_{\mathsf{ID},\theta,y}^{(\ell)} &:= g_2^{r_{\theta,\ell} y_2}.
\end{aligned}$$

Finally, $\mathcal{B}$ returns $\widetilde{\mathsf{SK}}_{\mathsf{ID}} := \{\widetilde{\mathsf{SK}}_{\mathsf{ID},\theta} := (\theta, \{\widetilde{\mathsf{SK}}_{\mathsf{ID},\theta}^{(\ell)}\}_{\ell=1}^{d})\}_{\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})}$.

When $\mathcal{A}$ submits $(M^*, \mathsf{ID}^*, \mathsf{T}^*)$, $\mathcal{B}$ picks $t, \delta_1^*, \delta_2^*, \delta_3^* \xleftarrow{\$} \mathbb{Z}_q$ and $R_{\mathrm{M}} \xleftarrow{\$} G_T$, and creates the challenge ciphertext $\widetilde{\mathsf{CT}}_{\mathsf{ID}^*,\mathsf{T}^*}^{*} := (\widetilde{\mathtt{ct}}_{\mathrm{M}}^{*}, \widetilde{\mathtt{ct}}_{x}^{*}, \widetilde{\mathtt{ct}}_{y}^{*}, \widetilde{\mathtt{ct}}_{\mathsf{ID}^*,\mathsf{T}^*}^{*}, \widetilde{\mathtt{tag}}^{*})$ as

$$\begin{aligned}
\widetilde{\mathtt{tag}}^{*} &:= \delta_1^* \mathsf{ID}^* + \delta_2^* \mathsf{T}^* + \delta_3^*,\\
\widetilde{\mathtt{ct}}_{\mathrm{M}}^{*} &:= R_{\mathrm{M}}, \ \ \widetilde{\mathtt{ct}}_{x}^{*} := (g_1^{\alpha})^t g_1^{c_2}, \ \ \widetilde{\mathtt{ct}}_{y}^{*} := g_1^t,\\
\widetilde{\mathtt{ct}}_{\mathsf{ID},\mathsf{T}}^{*} &:= \left(v^{\widetilde{\mathtt{tag}}^{*}} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t (g_1^{c_2})^{x\widetilde{\mathtt{tag}}^{*} + x_1 \mathsf{ID}^* + x_2 \mathsf{T}^*} Z.
\end{aligned}$$

Obviously, by setting $\mu := c_2$ the first component of the above ciphertext is random element of G_T and the others are the components of semi-functional ciphetext if $Z = g_1^{c_1 c_2}$. On the other hand, the above ciphertext consists of random elements of $\mathbf{G}_T \times \mathbf{G}_1^3 \times \mathbb{Z}_q$ if $Z = g_1^{c_1 c_2 + \eta}$ since we have

$$\begin{aligned}
\widetilde{\mathtt{ct}}_{\mathsf{ID}^*,\mathsf{T}^*}^{*} &:= \left(v^{\widetilde{\mathtt{tag}}^{*}} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t (g_1^{c_2})^{x\widetilde{\mathtt{tag}}^{*} + x_1 \mathsf{ID}^* + x_2 \mathsf{T}^*} Z\\
&= \left(v^{\widetilde{\mathtt{tag}}^{*}} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t (g_1^{c_2})^{x\widetilde{\mathtt{tag}}^{*} + x_1 \mathsf{ID}^* + x_2 \mathsf{T}^*} g_1^{x_3 \mu + \eta}\\
&= g_1^{\eta} \cdot \left(v^{\widetilde{\mathtt{tag}}^{*}} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t (g_1^{c_2})^{x\widetilde{\mathtt{tag}}^{*} + x_1 \mathsf{ID}^* + x_2 \mathsf{T}^* + x_3}.
\end{aligned}$$

This means that a random element η masks $\widetilde{\mathsf{ct}}^*_{\mathsf{ID}^*,\mathsf{T}^*}$. Since δ_1^*, δ_2^*, δ_3^*, c_2, and t are independent of each other, all elements of $\widetilde{\mathsf{CT}}_{\mathsf{ID}^*,\mathsf{T}^*}$ are independently chosen from $\mathcal{A}$'s view. □

We thus conclude the proof. □

6 Concluding Remarks

Human errors cannot be avoided, and therefore it is difficult to eliminate information leakage from our society. Decryption key exposure resistance (DKER) is one of the solutions for the key exposure problem in the context of RIBE. Since all the existing DKER RIBE schemes rely on the key re-randomization technique to achieve DKER, there are several open problems related to the limitation of the specific technique. In this paper, we newly introduced a mild form of DKER, called bounded DKER, to (partly) resolve the open problems. As a result, we proposed two RIBE constructions with bounded DKER, which are constructed from lattices and pairings, respectively. They are the first lattice-based/pairing-based anonymous RIBE scheme that achieves a kind of DKER. Our schemes are secure under the simple assumptions such as the LWE and SXDH assumptions.

It would be interesting to realize an anonymous RIBE scheme with (unbounded) DKER.

Acknowledgement. We would like to thank Shantian Cheng and Juanyang Zhang for their sincere discussion with us. We would like to thank Shuichi Katsumata for his helpful comments about lattice-based IBE.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010. 3, 10, 11, 28

[ABV+12] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Functional encryption for threshold functions (or fuzzy IBE) from lattices. In Marc Fischlin, Johannes A. Buchmann, and Mark Manulis, editors, *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 280–297. Springer, 2012. 28

[AFL16] Daniel Apon, Xiong Fan, and Feng-Hao Liu. Fully-secure lattice-based IBE as compact as PKE. *IACR Cryptology ePrint Archive*, 2016:125, 2016. 3, 28

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999. 10

[AP11] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory Comput. Syst.*, 48(3):535–553, 2011. 10

[BBDP01] MIhir Bellare, Alexandra Boldyreva, Anand Desai, David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer, 2001. 4

[BB11] Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *J. Cryptology*, 24(4):659–693, 2011. 3, 28

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. 3

[BGK08] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008*, pages 417–426. ACM, 2008. 3, 12

[BL16] Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and id-based encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security*, volume 10032 of *Lecture Notes in Computer Science*, pages 404–434, 2016. 3, 28

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13*, pages 575–584. ACM, 2013. 10, 11

[Boy10] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517. Springer, 2010. 3, 28

[BW06] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006. 3, 4, 8

[CCKS18] Donghoon Chang, Amit Kumar Chauhan, Sandeep Kumar, and Somitra Kumar Sanadhya. Revocable identity-based encryption from codes with rank metric. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018*, volume 10808 of *Lecture Notes in Computer Science*, pages 435–451. Springer, 2018. 3

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012. 3, 10, 28

[CLL+12a] Jie Chen, Hoon Wei Lim, San Ling, Le Su, and Huaxiong Wang. Anonymous and adaptively secure revocable IBE with constant size public parameters. *CoRR*, abs/1210.6441, 2012. 4, 9

[CLL+12b] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Khoa Nguyen. Revocable identity-based encryption from lattices. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *Information Security and Privacy - 17th Australasian Conference, ACISP 2012*, volume 7372 of *Lecture Notes in Computer Science*, pages 390–403. Springer, 2012. 3, 4, 6, 18, 21, 22, 23, 28

[CZ15] Shantian Cheng and Juanyang Zhang. Adaptive-id secure revocable identity-based encryption from lattices via subset difference method. In Javier Lopez and Yongdong Wu, editors, *Information Security Practice and Experience - 11th International Conference, ISPEC 2015*, volume 9065 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2015. 4, 27, 28

[EFF85] P. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51(1):79–89, 1985. 12

[ESY16] Keita Emura, Jae Hong Seo, and Taek-Young Youn. Semi-generic transformation of revocable hierarchical identity-based encryption and its DBDH instantiation. *IEICE Transactions*, 99-A(1):83–91, 2016. 3

[GLW12] Shafi Goldwasser, Allison B. Lewko, and David A. Wilson. Bounded-collusion IBE from key homomorphism. In Ronald Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012*, volume 7194 of *Lecture Notes in Computer Science*, pages 564–581. Springer, 2012. 4

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM, 2008. 3, 10, 28

[HK04] Swee-Huay Heng and Kaoru Kurosawa. k-resilient identity-based encryption in the standard model. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 67–80. Springer, 2004. 4

[HLCL18] Ziyuan Hu, Shengli Liu, Kefei Chen, and Joseph K. Liu. Revocable identity-based encryption from the computational Diffie-Hellman problem. In Willy Susilo and

Guomin Yang, editors, *Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings*, volume 10946 of *Lecture Notes in Computer Science*, pages 265–283. Springer, 2018. 3

[IWS15] Yuu Ishida, Yohei Watanabe, and Junji Shikata. Constructions of cca-secure revocable identity-based encryption. In Ernest Foo and Douglas Stebila, editors, *Information Security and Privacy - 20th Australasian Conference, ACISP 2015*, volume 9144 of *Lecture Notes in Computer Science*, pages 174–191. Springer, 2015. 3

[JR17] Charanjit S. Jutla and Arnab Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In *Journal of Cryptology*, volume 30, Issue 4, pages 1116–1156, 2017. 3, 7, 29, 36

[KMT18] Shuichi Katsumata, Takahiro Matsuda, and Atsushi Takayasu. Lattice-based revocable hierarchical identity-based encryption with decryption key exposure resistance. ePrint (to appear), 2018. 6, 13, 18, 27

[KRS99] Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. Coding constructions for blacklisting problems without computational assumptions. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference*, volume 1666 of *Lecture Notes in Computer Science*, pages 609–623. Springer, 1999. 12

[KY16] Shuichi Katsumata and Shota Yamada. Partitioning via non-linear polynomial functions: More compact ibes from ideal lattices and bilinear maps. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security*, volume 10032 of *Lecture Notes in Computer Science*, pages 682–712, 2016. 3, 28

[Lee16] Kwangsu Lee. Revocable hierarchical identity-based encryption with adaptive security. *IACR Cryptology ePrint Archive*, 2016:749, 2016. 3, 9, 33

[LLP17] Kwangsu Lee, Dong Hoon Lee, and Jong Hwan Park. Efficient revocable identity-based encryption via subset difference methods. *Des. Codes Cryptography*, 85(1):39–76, 2017. 3, 4

[LP16] Kwangsu Lee and Seunghwan Park. Revocable hierarchical identity-based encryption with shorter private keys and update keys. *IACR Cryptology ePrint Archive*, 2016:460, 2016. 3

[LV09] Benoît Libert and Damien Vergnaud. Adaptive-id secure revocable identity-based encryption. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2009. 3, 12

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012. 6, 10

[NNL01] Dalit Naor, Moni Naor, Jeff Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62, 2001. 3

[NWZ16] Khoa Nguyen, Huaxiong Wang, and Juanyang Zhang. Server-aided revocable identity-based encryption from lattices. In Sara Foresti and Giuseppe Persiano, editors, *Cryptology and Network Security - 15th International Conference, CANS 2016*, volume 10052 of *Lecture Notes in Computer Science*, pages 107–123, 2016. 4

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 333–342. ACM, 2009. 11

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM, 2005. 11

[RLPL15] Geumsook Ryu, Kwangsu Lee, Seunghwan Park, and Dong Hoon Lee. Unbounded hierarchical identity-based encryption with efficient revocation. In Howon Kim and Dooho Choi, editors, *Information Security Applications - 16th International Workshop, WISA 2015*, volume 9503 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2015. 3

[RS14] Somindu C. Ramanna and Palash Sarkar. Efficient (anonymous) compact HIBE from standard assumptions. In Sherman S. M. Chow, Joseph K. Liu, Lucas C. K. Hui, and Siu Ming Yiu, editors, *8th International Conference on Provable Security, ProvSec 2014*, volume 8782 of *Lecture Notes in Computer Science*, pages 243–258, 2014. 3, 7, 29, 36

[SW05] Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005. 3

[SE14b] Jae Hong Seo and Keita Emura. Revocable identity-based cryptosystem revisited: Security models and constructions. *IEEE Trans. Information Forensics and Security*, 9(7):1193–1205, 2014. 3, 4, 12, 28

[SE15] Jae Hong Seo and Keita Emura. Adaptive-ID Secure Revocable Hierarchical Identity-Based Encryption. In Keisuke Tanaka and Yuji Suga, editors, *Advances in Information*

and Computer Security - IWSEC 2015, volume 9241 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 1985. 9

[SE16] Jae Hong Seo and Keita Emura. Revocable hierarchical identity-based encryption via history-free approach. *Theor. Comput. Sci.*, 615:45–60, 2016. 3

[Sha84] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In George R. Blakley and David Chaum, editors, *Advances in Cryptology - CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1985. 3

[Sho06] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2006. 27

[TW17] Atsushi Takayasu and Yohei Watanabe. Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance. In Josef Pieprzyk and Suriadi Suriadi, editors, *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017*, volume 10342 of *Lecture Notes in Computer Science*, pages 184–204. Springer, 2017. 1, 5, 27

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005. 3, 28

[Wat09] Brent Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 24th Annual International Cryptology Conference*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009. 3, 8, 29

[WES17] Yohei Watanabe, Keita Emura, and Jae Hong Seo. New revocable IBE in prime-order groups: Adaptively secure, decryption key exposure resistant, and with short public parameters. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017*, volume 10159 of *Lecture Notes in Computer Science*, pages 432–449. Springer, 2017. 3, 4

[XWW+16] Qianqian Xing, Baosheng Wang, Xiaofeng Wang, Peixin Chen, Bo Yu, Yong Tang, and Xianming Gao. Unbounded revocable hierarchical identity-based encryption with adaptive-id security. In Jinjun Chen and Laurence T. Yang, editors, *18th IEEE International Conference on High Performance Computing and Communications; 14th IEEE International Conference on Smart City; 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, pages 430–437. IEEE, 2016. 9

[XWW+17] Qianqian Xing, Baosheng Wang, Xiaofeng Wang, Yong Tang, and Yi Wang. Déjà Q Encore RIBE: Anonymous Revocable Identity-Based Encryption with Short Parameters. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, IEEE, 2017. 4, 9

[XWWT18] Qianqian Xing, Baosheng Wang, Xiaofeng Wang, and Jing Tao. Unbounded and revocable hierarchical identity-based encryption with adaptive security, decryption key exposure resistant, and short public parameters. *PloS one*, 13(4):e0195204, 2018. 9

[Yam16] Shota Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 9666 of *Lecture Notes in Computer Science*, pages 32–62. Springer, 2016. 3, 28

[Yam17] Shota Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference*, volume 10403 of *Lecture Notes in Computer Science*, pages 161–193. Springer, 2017. 3, 28

[ZCZ16] Jiang Zhang, Yu Chen, and Zhenfeng Zhang. Programmable hash functions from lattices: Short signatures and ibes with small key sizes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference*, volume 9816 of *Lecture Notes in Computer Science*, pages 303–332. Springer, 2016. 3, 28

A Missing Proofs of Lemmas

A.1 Game 2_{j-1} → Game 2_j with $1 \leq j \leq Q_{sk}$

To show that the difference between **Game** 2_{j-1} and **Game** 2_j ($1 \leq j \leq Q_{sk}$) is negligible, we define *pseudo-normal secret keys* and *pseudo-semi-functional secret keys* as follows.

Pseudo-normal secret key for ID: Let η be a leaf node of $\mathtt{BT}$ assigned to ID, and parse a normal secret key $\mathsf{SK}_{\mathsf{ID}}$ as $\{\mathsf{SK}_{\mathsf{ID},\theta} = (\theta, \{\mathsf{SK}^{(\ell)}_{\mathsf{ID},\theta} = (\mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta}, \mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta,x}, \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,x}, \mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,x}, \mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta,y}, \mathtt{sk}'^{(\ell)}_{\mathsf{T},\theta,y}, \mathtt{sk}''^{(\ell)}_{\mathsf{T},\theta,y})\}_{\ell\in[d]})\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}$. Let $\delta_1^*, \delta_2^*, \delta_3^* \in \mathbb{Z}_q$ be elements of $\mathbb{Z}_q$ used for $\widetilde{\mathtt{tag}}^* := \delta_1^*\mathsf{ID}^* + \delta_2^*\mathsf{T}^* + \delta_3^*$ in the challenge ciphertext $\widetilde{\mathsf{CT}}^*_{\mathsf{ID}^*,\mathsf{T}^*}$. For each $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}})$ and $\ell \in [d]$, a pseudo-normal secret-key component $\overline{\mathsf{SK}}^{(\ell)}_{\mathsf{ID},\theta} := (\overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta}, \overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,y}, \overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,y}, \overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,y})$ is computed by

$$
\begin{aligned}
\overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta} &:= \mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta},\\
\overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,x} &:= \mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta,x} \cdot g_2^{-\gamma_{\theta,\ell}} = g_2^{r_{\theta,\ell}x-\gamma_{\theta,\ell}},\\
\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,x} &:= \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,x} \cdot g_2^{\gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}+\delta_3^*)} = P^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)+\gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}+\delta_3^*)},\\
\overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,x} &:= \mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,x} \cdot g_2^{\delta_2^*} = g_2^{r_{\theta,\ell}x_2+\delta_2^*},\\
\overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,y} &:= \mathtt{sk}^{(\ell)}_{\mathsf{ID},\theta,y} \cdot g_2^{\alpha\gamma_{\theta,\ell}} = g_2^{r_{\theta,\ell}y+\alpha\gamma_{\theta,\ell}},
\end{aligned}
$$

$$\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,y} := \mathtt{sk}'^{(\ell)}_{\mathsf{ID},\theta,y} \cdot g_2^{-\alpha\gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}+\delta_3^*)} = P^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)-\alpha\gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}+\delta_3^*)},$$
$$\overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,y} := \mathtt{sk}''^{(\ell)}_{\mathsf{ID},\theta,y} \cdot g_2^{-\alpha\delta_2^*} = g_2^{r_{\theta,\ell}y_2-\alpha\delta_2^*},$$

where $\gamma_{\theta,\ell} \xleftarrow{\$} \mathbb{Z}_q$. A pseudo-normal secret key is $\overline{\mathsf{SK}}_{\mathsf{ID}} := \{\overline{\mathsf{SK}}_{\mathsf{ID},\theta} = (\theta, \{\overline{\mathsf{SK}}^{(\ell)}_{\mathsf{ID},\theta}\}_{\ell\in[d]})\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}$.

Pseudo-semi-functional secret key for ID: Let η be a leaf node of $\mathtt{BT}$ assigned to ID, and parse a pseudo-normal secret key $\overline{\mathsf{SK}}_{\mathsf{ID}}$ as $\{\overline{\mathsf{SK}}_{\mathsf{ID},\theta} = \{(\theta, \overline{\mathsf{SK}}^{(\ell)}_{\mathsf{ID},\theta} = (\overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta}, \overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,y}, \overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,y}, \overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,y})\}_{\ell\in[d]})\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}$. For each $\theta \in \mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})$ and $\ell \in [d]$, a pseudo-semi-functional secret-key component $\widehat{\mathsf{SK}}^{(\ell)}_{\mathsf{ID},\theta} := (\overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta}, \overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,x}, \widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,x}, \overline{\mathtt{sk}}^{(\ell)}_{\mathsf{ID},\theta,y}, \widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,y}, \overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID},\theta,y})$ is computed by

$$\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,x} := \overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,x} \cdot g_2^{\phi_{\theta,\ell}} = P^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}+x_3)+\phi_{\theta,\ell}+\gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}+\delta_3^*)},$$
$$\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,y} := \overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID},\theta,y} \cdot g_2^{-\alpha\phi_{\theta,\ell}} = P^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}+y_3)-\alpha(\phi_{\theta,\ell}+\gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}+\delta_3^*))},$$

where $\phi_{\theta,\ell} \xleftarrow{\$} \mathbb{Z}_q$. A pseudo-semi-functional secret key is $\widehat{\mathsf{SK}}_{\mathsf{ID}} := \{\widehat{\mathsf{SK}}_{\mathsf{ID},\theta} = \{(\theta, \widehat{\mathsf{SK}}^{(\ell)}_{\mathsf{ID},\theta}\}_{\ell\in[d]})\}_{\theta\in\mathsf{Path}(\mathtt{BT},\eta_{\mathsf{ID}})}$.

Note that g_2^α is needed to compute pseudo-normal/pseudo-semi-functional keys, and that normal ciphetexts can be decrypted by decryption keys generated from pseudo-normal/pseudo-semi-functional secret keys and normal/semi-functional key updates.

We additionally define the following games.

Game $2_{j,1}$ $(0 \le j \le Q_{sk}-1)$**:** The game is the same as **Game** 2_j except that for a $(j+1)$-st secret key reveal query ID_{j+1}, a pseudo-normal secret key $\overline{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$ is returned.

Game $2_{j,2}$ $(0 \le j \le Q_{sk}-1)$**:** The game is the same as **Game** $2_{j,1}$ except that for $(j+1)$-st secret key reveal query ID_{j+1}, a pseudo-semi-functional secret key $\widehat{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$ is returned.

We show that for every $j \in \{0,1,\ldots,Q_{sk}-1\}$,

$$\left|\Pr[E_{2_j}] - \Pr[E_{2_{j+1}}]\right| \le \left|\Pr[E_{2_j}] - \Pr[E_{2_{j,1}}]\right| + \left|\Pr[E_{2_{j,1}}] - \Pr[E_{2_{j,2}}]\right| + \left|\Pr[E_{2_{j,2}}] - \Pr[E_{2_{j+1}}]\right|,$$

is negligible.

Lemma 12. *For every $j \in \{0,1,\ldots,Q_{sk}-1\}$, if there exists a PPT adversary $\mathcal{A}$ to distinguish* **Game** 2_j *and* **Game** $2_{j,1}$*, then there exists a PPT adversary $\mathcal{B}$ to break the DDH2 assumption.*

Proof. Given the DDH2 instance $(\mathbb{G}, g_2^{c_1}, g_2^{c_2})$ with Z. The simulator $\mathcal{B}$ uses $\mathcal{A}$ to distinguish if $\mathbb{Z}_q$ is distributed as $g_2^{c_1c_2}$ or $g_2^{c_1c_2+\gamma}$.

$\mathcal{B}$ chooses $\alpha, x', x_0, x'_1, x'_2, x'_3, y', y_0, y'_1, y'_2, y'_3, \delta_1^*, \delta_2^*, \delta_3^* \xleftarrow{\$} \mathbb{Z}_q$, and implicitly sets

$$x := x' - c_2, \quad y := y' + c_2\alpha, \quad x_1 := x'_1 + c_2\delta_1^*, \quad y_1 := y'_1 - c_2\delta_1^*\alpha,$$

$$x_2 := x_2' + c_2\delta_2^*, \quad y_2 := y_2' - c_2\delta_2^*\alpha, \quad x_3 := x_3' + c_2\delta_3^*, \quad y_3 := y_3' - c_2\delta_3^*\alpha.$$

Note that $x_1, x_2, x_3, y_1, y_2, y_3$ do not leak any information of $\delta_1^*, \delta_2^*, \delta_3^*$ since $x_1', x_2', x_3', y_1', y_2', y_3'$ and c_2 mask them in the information-theoretic sense. Therefore, although $\delta_1^*, \delta_2^*, \delta_3^*$ will be used for the challenge ciphertext, choosing them at this point does not affect any distribution of them. $\mathcal{B}$ then computes

$$v := g_1^{x'\alpha+y'}, \quad z := e(g_1, g_2)^{x_0\alpha+y_0}, \quad u_{\mathrm{ID}} := g_1^{x_1'\alpha+y_1'}, \quad u_{\mathrm{T}} := g_1^{x_2'\alpha+y_2'}, \quad h := g_1^{x_3'\alpha+y_3'}.$$

$\mathcal{B}$ runs $\mathtt{BT} \leftarrow \mathsf{CS.SetUp}(N)$ by arbitrarily choosing $N \in \mathbb{N}$, and sends $\mathsf{PP} := (\mathbb{G}, g_1^\alpha, v, u_{\mathrm{ID}}, u_{\mathrm{T}}, h, z)$ to $\mathcal{A}$. Note that $\mathcal{B}$ does not know all the master key. However, $\mathcal{B}$ can compute

$$\begin{aligned}
&g_2^x := g_2^{x'}(g_2^{c_2})^{-1}, \quad g_2^x := g_2^{y'}(g_2^{c_2})^{\alpha}, \quad g_2^{x_1} := g_2^{x_1'}(g_2^{c_2})^{\delta_1^*}, \quad g_2^{y_1} := g_2^{y_1'}(g_2^{c_2})^{-\delta_1^*\alpha},\\
&g_2^{x_2} := g_2^{x_2'}(g_2^{c_2})^{\delta_2^*}, \quad g_2^{y_2} := g_2^{y_2'}(g_2^{c_2})^{-\delta_2^*\alpha}, \quad g_2^{x_3} := g_2^{x_3'}(g_2^{c_2})^{\delta_3^*}, \quad g_2^{y_3} := g_2^{y_3'}(g_2^{c_2})^{-\delta_3^*\alpha}.
\end{aligned}$$

Therefore, $\mathcal{B}$ can compute any secret keys and key updates in any forms (i.e., normal ones and semi-functional ones).

We show how $\mathcal{B}$ responds to a $(j+1)$-st secret key reveal query ID_{j+1}. $\mathcal{B}$ first runs $(\eta_{\mathsf{ID}_{j+1}}, \mathtt{BT}) \leftarrow \mathsf{CS.Assign}(\mathtt{BT}, \mathsf{ID}_{j+1})$. For each node $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}_{j+1}})$, $\mathcal{B}$ recalls $\{P_{\theta,x}^{(\ell)}, P_{\theta,y}^{(\ell)}\}_{\ell=1}^d$ if they were defined. Otherwise, $\mathcal{B}$ chooses them from G_2 randomly, and stores them in θ. For every $\ell \in [d]$, $\mathcal{B}$ chooses $r_{\theta,\ell}^{(1)}, r_{\theta,\ell}^{(2)} \xleftarrow{\$} \mathbb{Z}_q$ and computes

$$\begin{aligned}
\mathtt{sk}_{\mathsf{ID}_{j+1},\theta}^{(\ell)} &:= (g_2^{c_1})^{r_{\theta,\ell}^{(1)}} \cdot g_2^{r_{\theta,\ell}^{(2)}},\\
\mathtt{sk}_{\mathsf{ID}_{j+1},\theta,x}^{(\ell)} &:= g_2^{r_{\theta,\ell}^{(2)}x'} \cdot (g_2^{c_1})^{r_{\theta,\ell}^{(1)}x'} \cdot (g_2^{c_2})^{-r_{\theta,\ell}^{(2)}} \cdot Z^{-r_{\theta,\ell}^{(1)}},\\
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} &:= P_{\theta,x}^{(\ell)} \cdot g_2^{r_{\theta,\ell}^{(2)}(x_1'\mathsf{ID}_{j+1}+x_3')} \cdot (g_2^{c_1})^{r_{\theta,\ell}^{(1)}(x_1'\mathsf{ID}_{j+1}+x_3')} \cdot (g_2^{c_2})^{r_{\theta,\ell}^{(2)}(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)} \cdot Z^{r_{\theta,\ell}^{(1)}(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)},\\
\mathtt{sk}''^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} &:= g_2^{r_{\theta,\ell}^{(2)}x_2'} \cdot (g_2^{c_1})^{r_{\theta,\ell}^{(1)}x_2'} \cdot (g_2^{c_2})^{r_{\theta,\ell}^{(2)}\delta_2^*} \cdot Z^{r_{\theta,\ell}^{(1)}\delta_2^*},\\
\mathtt{sk}_{\mathsf{ID}_{j+1},\theta,y}^{(\ell)} &:= g_2^{r_{\theta,\ell}^{(2)}y'} \cdot (g_2^{c_1})^{r_{\theta,\ell}^{(1)}y'} \cdot (g_2^{c_2})^{r_{\theta,\ell}^{(2)}\alpha} \cdot Z^{r_{\theta,\ell}^{(1)}\alpha},\\
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &:= P_{\theta,y}^{(\ell)} \cdot g_2^{r_{\theta,\ell}^{(2)}(y_1'\mathsf{ID}_{j+1}+y_3')} \cdot (g_2^{c_1})^{r_{\theta,\ell}^{(1)}(y_1'\mathsf{ID}_{j+1}+y_3')} \cdot (g_2^{c_2})^{-r_{\theta,\ell}^{(2)}\alpha(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)} \cdot Z^{-r_{\theta,\ell}^{(1)}\alpha(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)},\\
\mathtt{sk}''^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &:= g_2^{r_{\theta,\ell}^{(2)}y_2'} \cdot (g_2^{c_1})^{r_{\theta,\ell}^{(1)}y_2'} \cdot (g_2^{c_2})^{-r_{\theta,\ell}^{(2)}\alpha\delta_2^*} \cdot Z^{-r_{\theta,\ell}^{(1)}\alpha\delta_2^*}.
\end{aligned}$$

The above secret key is normal if $Z = g_2^{c_1c_2}$ since we have

$$\begin{aligned}
\mathtt{sk}_{\mathsf{ID}_{j+1},\theta}^{(\ell)} &= g_2^{r_{\theta,\ell}^{(1)}c_1+r_{\theta,\ell}^{(2)}} = g_2^{r_{\theta,r}},\\
\mathtt{sk}_{\mathsf{ID}_{j+1},\theta,x}^{(\ell)} &= g_2^{(r_{\theta,\ell}^{(1)}c_1+r_{\theta,\ell}^{(2)})(x'-c_2)} = g_2^{r_{\theta,\ell}x},
\end{aligned}$$

$$
\begin{aligned}
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} &= P^{(\ell)}_{\theta,x} \cdot g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})\left((x'_1+c_2\delta^*_1)\mathsf{ID}_{j+1}+(x'_3+c_2\delta^*_3)\right)} = P^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}_{j+1}+x_3)},\\
\mathtt{sk}''^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} &= g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})(x'_2+c_2\delta^*_2)} = g_2^{r_{\theta,\ell}x_2},\\
\mathtt{sk}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &= g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})(y'+\alpha c_2)} = g_2^{r_{\theta,\ell}y},\\
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &= P^{(\ell)}_{\theta,y} \cdot g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})\left((y'_1-c_2\alpha\delta^*_1)\mathsf{ID}_{j+1}+(y'_3-c_2\alpha\delta^*_3)\right)} = P^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}_{j+1}+y_3)},\\
\mathtt{sk}''^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &= g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})(y'_2-c_2\alpha\delta^*_2)} = g_2^{r_{\theta,\ell}y_2},
\end{aligned}
$$

where $r_{\theta,\ell} := r^{(1)}_{\theta,\ell}c_1 + r^{(2)}_{\theta,\ell}$.

On the other hand, the above key is pseudo-normal if $Z = g_2^{c_1c_2+\gamma}$ since we have

$$
\begin{aligned}
\mathtt{sk}^{(\ell)}_{\mathsf{ID}_{j+1},\theta} &= g_2^{r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell}} = g_2^{r_{\theta,r}},\\
\mathtt{sk}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} &= g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})(x'-c_2)-\gamma r^{(1)}_{\theta,\ell}} = g_2^{r_{\theta,\ell}x-\gamma_{\theta,\ell}},\\
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} &= P^{(\ell)}_{\theta,x} \cdot g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})\left((x'_1+c_2\delta^*_1)\mathsf{ID}_{j+1}+(x'_3+c_2\delta^*_3)\right)+\gamma r^{(1)}_{\theta,\ell}(\delta^*_1\mathsf{ID}_{j+1}+\delta^*_3)}\\
&= P^{(\ell)}_{\theta,x} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}_{j+1}+x_3)+\gamma_{\theta,\ell}(\delta^*_1\mathsf{ID}_{j+1}+\delta^*_3)},\\
\mathtt{sk}''^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} &= g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})(x'_2+c_2\delta^*_2)+\gamma r^{(1)}_{\theta,\ell}\delta^*_2} = g_2^{r_{\theta,\ell}x_2+\gamma_{\theta,\ell}\delta^*_2},\\
\mathtt{sk}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &= g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})(y'+\alpha c_2)+\gamma r^{(1)}_{\theta,\ell}\alpha} = g_2^{r_{\theta,\ell}y+\gamma_{\theta,\ell}\alpha},\\
\mathtt{sk}'^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &= P^{(\ell)}_{\theta,y} \cdot g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})\left((y'_1-c_2\alpha\delta^*_1)\mathsf{ID}_{j+1}+(y'_3-c_2\alpha\delta^*_3)\right)-\gamma r^{(1)}_{\theta,\ell}\alpha(\delta^*_1\mathsf{ID}_{j+1}+\delta^*_3)}\\
&= P^{(\ell)}_{\theta,y} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}_{j+1}+y_3)-\gamma\theta,\ell\alpha(\delta^*_1\mathsf{ID}_{j+1}+\delta^*_3)},\\
\mathtt{sk}''^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} &= g_2^{(r^{(1)}_{\theta,\ell}c_1+r^{(2)}_{\theta,\ell})(y'_2-c_2\alpha\delta^*_2)-\gamma r^{(1)}_{\theta,\ell}\alpha\delta^*_2} = g_2^{r_{\theta,\ell}y_2-\gamma_{\theta,\ell}\alpha\delta^*_2},
\end{aligned}
$$

where $r_{\theta,\ell} := r^{(1)}_{\theta,\ell}c_1 + r^{(2)}_{\theta,\ell}$ and $\gamma_{\theta,\ell} := \gamma r^{(1)}_{\theta,\ell}$. It is obvious that $r_{\theta,\ell}$ and $\gamma_{\theta,\ell}$ is independent of each other from the view point of $\mathcal{A}$.

When $\mathcal{A}$ submits $(M^*, \mathsf{ID}^*, \mathsf{T}^*)$, $\mathcal{B}$ picks $t, \mu \xleftarrow{\$} \mathbb{Z}_q$ and creates the challenge ciphertext $\widetilde{\mathsf{CT}}^*_{\mathsf{ID}^*,\mathsf{T}^*} := (\widetilde{\mathtt{ct}}^*_{\mathrm{M}}, \widetilde{\mathtt{ct}}^*_x, \widetilde{\mathtt{ct}}^*_y, \widetilde{\mathtt{ct}}^*_{\mathsf{ID},\mathsf{T}}, \widetilde{\mathtt{tag}}^*)$ as

$$
\begin{aligned}
\widetilde{\mathtt{ct}}^*_{\mathrm{M}} &:= M^* \cdot z^t \cdot e(g_1, g_2)^{x_0\mu},\\
\widetilde{\mathtt{ct}}^*_x &:= (g_1^{\alpha})^t \cdot g_1^{\mu},\\
\widetilde{\mathtt{ct}}^*_y &:= g_1^t,\\
\widetilde{\mathtt{ct}}^*_{\mathsf{ID}^*,\mathsf{T}^*} &:= \left(v^{\widetilde{\mathtt{tag}}^*} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t \cdot g_1^{\mu(x'\mathtt{tag}^*+x'_1\mathsf{ID}^*+x'_2\mathsf{T}^*+x'_3)}
\end{aligned}
$$

$$= \left(v^{\widetilde{\mathtt{tag}}^*} u_{\mathrm{ID}}^{\mathsf{ID}^*} u_{\mathrm{T}}^{\mathsf{T}^*} h\right)^t \cdot g_1^{\mu((x'-c_2)\widetilde{\mathtt{tag}}^* + (x'_1+c_2\delta_1^*)\mathsf{ID}^* + (x'_2+c_2\delta_2^*)\mathsf{T}^* + x'_3+c_2\delta_3^*)} \cdot g_1^{\mu c_2(\widetilde{\mathtt{tag}}^* - \delta_1\mathsf{ID}^* - \delta_2\mathsf{T}^* - \delta_3)}$$
$$= \left(w^{\widetilde{\mathtt{tag}}^*} u^{\mathsf{ID}^*} h v^{\mathsf{T}^*}\right)^t \cdot g_1^{\mu(x\widetilde{\mathtt{tag}}^* + x_1\mathsf{ID}^* + x_2\mathsf{T}^* + x_3)},$$

where $\widetilde{\mathtt{tag}}^* := \delta_1^*\mathsf{ID}^* + \delta_2^*\mathsf{T}^* + \delta_3^*$. □

Lemma 13. *For every $j \in \{0, 1, \ldots, Q_{sk}-1\}$, all of the distributions in* **Game** $2_{j,1}$ *are identical to those in* **Game** $2_{j,2}$ *from the view point of any PPT adversary $\mathcal{A}$. Namely, it holds* $\Pr[E_{2_{j,1}}] = \Pr[E_{2_{j,2}}]$ *for every* $j \in \{0, 1, \ldots, Q_{\mathsf{sk}}-1\}$.

Proof. The difference between between **Game** $2_{j,1}$ and **Game** $2_{j,2}$ is only that a secret key for the $(j+1)$-st query is pseudo-normal $\overline{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$ or pseudo-semi-functional $\widehat{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$. More specifically, the difference between $\overline{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$ and $\widehat{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$ is whether or not the component $\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x}$ and $\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y}$ (or $\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x}$ and $\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y}$) contain $\phi_{\theta,\ell}$ for all $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}_{j+1}})$ and $\ell \in [d]$, where $\eta_{\mathsf{ID}_{j+1}}$ is a leaf node corresponding to ID_{j+1}.

For every $\theta \in \mathsf{Path}(\mathtt{BT}, \eta_{\mathsf{ID}_{j+1}})$ and $\ell \in [d]$, the corresponding components of $\overline{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$ are:

$$\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} = P_{\theta,x}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}_{j+1}+x_3) + \gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)},$$
$$\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} = P_{\theta,y}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}_{j+1}+y_3) - \gamma_{\theta,\ell}\alpha(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)}.$$

For any $\mathsf{T} \in \mathcal{T}$, $\mathcal{A}$ can compute

$$\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} \cdot \left(\overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x}\right)^{\mathsf{T}} = P_{\theta,x}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}_{j+1}+x_2\mathsf{T}+x_3) + \gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}_{j+1}+\delta_2^*\mathsf{T}+\delta_3^*)},$$
$$\overline{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} \cdot \left(\overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y}\right)^{\mathsf{T}} = P_{\theta,y}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}_{j+1}+y_2\mathsf{T}+y_3) - \gamma_{\theta,\ell}\alpha(\delta_1^*\mathsf{ID}_{j+1}+\delta_2^*\mathsf{T}+\delta_3^*)},$$

where $\overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} = g_2^{r_{\theta,\ell}x_2+\delta_2^*}$ and $\overline{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} = g_2^{r_{\theta,\ell}y_2-\alpha\delta_2^*}$.

On the other hand, the corresponding components of $\widehat{\mathsf{SK}}_{\mathsf{ID}_{j+1}}$ are:

$$\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} = P_{\theta,x}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}_{j+1}+x_3) + \phi_{\theta,\ell} + \gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)},$$
$$\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} = P_{\theta,y}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}_{j+1}+y_3) - \alpha\phi_{\theta,\ell} - \gamma_{\theta,\ell}\alpha(\delta_1^*\mathsf{ID}_{j+1}+\delta_3^*)}.$$

For any $\mathsf{T} \in \mathcal{T}$, $\mathcal{A}$ can compute

$$\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} \cdot \left(\widehat{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x}\right)^{\mathsf{T}} = P_{\theta,x}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(x_1\mathsf{ID}_{j+1}+x_2\mathsf{T}+x_3) + \gamma_{\theta,\ell}(\delta_1^*\mathsf{ID}_{j+1}+\delta_2^*\mathsf{T}+\delta_3^*+\frac{\phi_{\theta,\ell}}{\gamma_{\theta,\ell}})},$$
$$\widehat{\mathtt{sk}'}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} \cdot \left(\widehat{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y}\right)^{\mathsf{T}} = P_{\theta,y}^{(\ell)} \cdot g_2^{r_{\theta,\ell}(y_1\mathsf{ID}_{j+1}+y_2\mathsf{T}+y_3) - \gamma_{\theta,\ell}\alpha(\delta_1^*\mathsf{ID}_{j+1}+\delta_2^*\mathsf{T}+\delta_3^*+\frac{\phi_{\theta,\ell}}{\gamma_{\theta,\ell}})},$$

where $\widehat{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,x} = g_2^{r_{\theta,\ell}x_2+\delta_2^*}$ and $\widehat{\mathtt{sk}''}^{(\ell)}_{\mathsf{ID}_{j+1},\theta,y} = g_2^{r_{\theta,\ell}y_2-\alpha\delta_2^*}$.

The tag for the challenge ciphertext in both games is $\widetilde{\mathsf{tag}}^* := \delta_1^*\mathsf{ID}^* + \delta_2^*\mathsf{T}^* + \delta_3^*$. We can write

$$\begin{pmatrix} \widetilde{\mathsf{tag}}^* \\ \delta_1^*\mathsf{ID}_{j+1} + \delta_2^*\mathsf{T} + \delta_3^* \end{pmatrix} = \underbrace{\begin{pmatrix} \mathsf{ID}^* & \mathsf{T}^* & 1 \\ \mathsf{ID}_{j+1} & \mathsf{T}_j & 1 \end{pmatrix}}_{(a)} \begin{pmatrix} \delta_1^* \\ \delta_2^* \\ \delta_3^* \end{pmatrix}.$$

Since $\mathsf{ID}_{j+1} \neq \mathsf{ID}^*$, the matrix (a) has full rank. Namely, in **Game** $2_{j,1}$, $\widetilde{\mathsf{tag}}^*$ and $\delta_1^*\mathsf{ID}_{j+1} + \delta_2^*\mathsf{T} + \delta_3^*$ are independently distributed from $\mathcal{A}$'s view. Moreover, we can write

$$\begin{pmatrix} \widetilde{\mathsf{tag}}^* \\ \delta_1^*\mathsf{ID}_{j+1} + \delta_2^*\mathsf{T} + \delta_3^* + \frac{\phi_{\theta,\ell}}{\gamma_{\theta,\ell}} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathsf{ID}^* & \mathsf{T}^* & 1 \\ \mathsf{ID}_{j+1} & \mathsf{T}_j & 1 + \frac{\phi_{\theta,\ell}}{\delta_3^*\gamma_{\theta,\ell}} \end{pmatrix}}_{(b)} \begin{pmatrix} \delta_1^* \\ \delta_2^* \\ \delta_3^* \end{pmatrix}.$$

Since the matrix (b) obviously has full rank, in **Game** $2_{j,2}$, $\widetilde{\mathsf{tag}}^*$ and $\delta_1^*\mathsf{ID}_{j+1} + \delta_2^*\mathsf{T} + \delta_3^* + \phi_{\theta,\ell}/\gamma_{\theta,\ell}$ are independently distributed from $\mathcal{A}$'s view.

Hence, for every $\ell \in [d]$, the following distributions

$$\left\{\widetilde{\mathsf{tag}}^*,\ \delta_1^*\mathsf{ID}_{j+1} + \delta_2^*\mathsf{T} + \delta_3^*\right\} \quad \text{and} \quad \left\{\widetilde{\mathsf{tag}}^*,\ \delta_1^*\mathsf{ID}_{j+1} + \delta_2^*\mathsf{T} + \delta_3^* + \frac{\phi_{\theta,\ell}}{\gamma_{\theta,\ell}}\right\}$$

are equivalent from $\mathcal{A}$'s view since $\phi_{\theta,\ell}$ and $\gamma_{\theta,\ell}$ are randomly chosen. Note that $\gamma_{\theta,\ell} \neq 0$ since $\gamma_{\theta,\ell} = 0$ means that the $(j+1)$-st secret key is normal or semi-functional, not pseudo-normal or pseudo-semi-functional. □

Lemma 14. *For every $j \in \{0, 1, \ldots, Q_{\mathsf{sk}} - 1\}$, if there exists a PPT adversary $\mathcal{A}$ to distinguish* **Game** $2_{j,2}$ *and* **Game** 2_{j+1}*, then there exists a PPT adversary $\mathcal{B}$ to break the DDH2 assumption.*

Proof. We omit the proof since this lemma can be proved in a similar way to Lemma 12. □

A.2 Game 3_{j-1} → Game 3_j with $1 \leq j \leq |\mathcal{T}|$

To show that the difference between **Game** 3_{j-1} and **Game** 3_j $(1 \leq j \leq |\mathcal{T}|)$ is negligible, we define *pseudo-normal key updates* and *pseudo-semi-functional key updates* as well as secret keys.

Pseudo-normal key update for T: Parse a normal key update KU_T as $(\{\mathsf{KU}_{\mathsf{T},\theta} = (\theta, \mathtt{ku}_{\mathsf{T},\theta}, \mathtt{ku}_{\mathsf{T},\theta,x}, \mathtt{ku}'_{\mathsf{T},\theta,x}, \mathtt{ku}''_{\mathsf{T},\theta,x}, \mathtt{ku}_{\mathsf{T},\theta,y}, \mathtt{ku}'_{\mathsf{T},\theta,y}, \mathtt{ku}''_{\mathsf{T},\theta,y})\}_{\theta \in \mathsf{KUNode}(\mathsf{BT},\mathsf{RL}_\mathsf{T})}, \mathcal{F}_\mathsf{T})$. Let $\delta_1^*, \delta_2^*, \delta_3^* \in \mathbb{Z}_q$ be elements of $\mathbb{Z}_q$ used for $\widetilde{\mathsf{tag}}^* := \delta_1^*\mathsf{ID}^* + \delta_2^*\mathsf{T}^* + \delta_3^*$ in the challenge ciphertext $\widetilde{\mathsf{CT}}^*_{\mathsf{ID}^*,\mathsf{T}^*}$. For each $\theta \in \mathsf{KUNode}(\mathsf{BT}, \mathsf{RL}_\mathsf{T})$, a pseudo-normal key-update component $\overline{\mathsf{KU}}_{\mathsf{T},\theta} := (\theta, \overline{\mathtt{ku}}_{\mathsf{T},\theta}, \overline{\mathtt{ku}}_{\mathsf{T},\theta,x}, \overline{\mathtt{ku}'}_{\mathsf{T},\theta,x}, \overline{\mathtt{ku}''}_{\mathsf{T},\theta,x}, \overline{\mathtt{ku}}_{\mathsf{T},\theta,y}, \overline{\mathtt{ku}'}_{\mathsf{T},\theta,y}, \overline{\mathtt{ku}''}_{\mathsf{T},\theta,y})$ is computed by

$$\begin{aligned} \overline{\mathtt{ku}}_{\mathsf{T},\theta} &:= \mathtt{ku}_{\mathsf{T},\theta}, \\ \overline{\mathtt{ku}}_{\mathsf{T},\theta,x} &:= \mathtt{ku}_{\mathsf{T},\theta,x} g_2^{-\gamma_\theta} = g_2^{s_\theta x - \gamma_\theta}, \end{aligned}$$

$$\overline{\mathsf{ku}'}_{\mathsf{T},\theta,x} := \mathsf{ku}'_{\mathsf{T},\theta,x} \cdot g_2^{\gamma_\theta(\delta_2^*\mathsf{T}+\delta_3^*)} = \left(\prod_{\ell\in\mathcal{F}_\mathsf{T}} \left(P_{\theta,x}^{(\ell)}\right)^{-1}\right)^{-1} \cdot g_2^{x_0+s_\theta(x_2\mathsf{T}+x_3)+\gamma_\theta(\delta_2^*\mathsf{T}+\delta_3^*)},$$
$$\overline{\mathsf{ku}''}_{\mathsf{T},\theta,x} := \mathsf{ku}''_{\mathsf{T},\theta,x} g_2^{\delta_1^*} = g_2^{s_{\theta,\ell}x_1+\delta_1^*},$$
$$\overline{\mathsf{ku}}_{\mathsf{T},\theta,y} := \mathsf{ku}_{\mathsf{T},\theta,y} g_2^{\alpha\gamma_\theta} = g_2^{s_\theta y+\alpha\gamma_\theta},$$
$$\overline{\mathsf{ku}'}_{\mathsf{T},\theta,y} := \mathsf{ku}'_{\mathsf{T},\theta,y} \cdot g_2^{-\alpha\gamma_\theta(\delta_2^*\mathsf{T}+\delta_3^*)} = \left(\prod_{\ell\in\mathcal{F}_\mathsf{T}} \left(P_{\theta,y}^{(\ell)}\right)^{-1}\right)^{-1} \cdot g_2^{y_0+s_\theta(y_2\mathsf{T}+y_3)-\alpha\gamma_\theta(\delta_2^*\mathsf{T}+\delta_3^*)},$$
$$\overline{\mathsf{ku}''}_{\mathsf{T},\theta,y} := \mathsf{ku}''_{\mathsf{T},\theta,y} g_2^{-\alpha\delta_1^*} = g_2^{s_\theta y_1-\alpha\delta_1^*},$$

where $\gamma_\theta \xleftarrow{\$} \mathbb{Z}_q$. A pseudo-normal key update is $\overline{\mathsf{KU}}_\mathsf{T} := (\{\overline{\mathsf{ku}}_{\mathsf{T},\theta}\}_{\theta\in\mathsf{KUNode}(\mathsf{BT},\mathsf{RL}_\mathsf{T})}, \mathcal{F}_\mathsf{T})$.

Pseudo-semi-functional key update for T: Parse a pseudo-normal key update $\overline{\mathsf{KU}}_\mathsf{T}$ as $(\{\overline{\mathsf{KU}}_{\mathsf{T},\theta} = (\theta, \overline{\mathsf{ku}}_{\mathsf{T},\theta}, \overline{\mathsf{ku}}_{\mathsf{T},\theta,x}, \overline{\mathsf{ku}'}_{\mathsf{T},\theta,x}, \overline{\mathsf{ku}''}_{\mathsf{T},\theta,x}, \overline{\mathsf{ku}}_{\mathsf{T},\theta,y}, \overline{\mathsf{ku}'}_{\mathsf{T},\theta,y}, \overline{\mathsf{ku}''}_{\mathsf{T},\theta,y})\}_{\theta\in\mathsf{KUNode}(\mathsf{BT},\mathsf{RL}_\mathsf{T})}, \mathcal{F}_\mathsf{T})$. For each $\theta \in \mathsf{KUNode}(\mathsf{BT},\mathsf{RL}_\mathsf{T})$, a pseudo-semi-functional key-update component $\widehat{\mathsf{KU}}_{\mathsf{T},\theta} := (\theta, \overline{\mathsf{ku}}_{\mathsf{T},\theta}, \overline{\mathsf{ku}}_{\mathsf{T},\theta,x}, \widehat{\mathsf{ku}'}_{\mathsf{T},\theta,x}, \overline{\mathsf{ku}''}_{\mathsf{T},\theta,x}, \overline{\mathsf{ku}}_{\mathsf{T},\theta,y}, \widehat{\mathsf{ku}'}_{\mathsf{T},\theta,y}, \overline{\mathsf{ku}''}_{\mathsf{T},\theta,y})$ is computed by

$$\widehat{\mathsf{ku}'}_{\mathsf{T},\theta,x} := \overline{\mathsf{ku}'}_{\mathsf{T},\theta,x} \cdot g_2^{\psi_\theta} = \left(\prod_{\ell\in\mathcal{F}_\mathsf{T}} \left(P_{\theta,x}^{(\ell)}\right)^{-1}\right)^{-1} \cdot g_2^{x_0+s_\theta(x_2\mathsf{T}+x_3)+\psi_\theta+\gamma_\theta(\delta_2^*\mathsf{T}+\delta_3^*)},$$
$$\widehat{\mathsf{ku}'}_{\mathsf{T},\theta,y} := \overline{\mathsf{ku}'}_{\mathsf{T},\theta,y} \cdot g_2^{-\alpha\psi_\theta} = \left(\prod_{\ell\in\mathcal{F}_\mathsf{T}} \left(P_{\theta,y}^{(\ell)}\right)^{-1}\right)^{-1} \cdot g_2^{y_0+s_\theta(y_2\mathsf{T}+y_3)-\alpha(\psi_\theta+\gamma_\theta(\delta_2^*\mathsf{T}+\delta_3^*))},$$

where $\psi_\theta \xleftarrow{\$} \mathbb{Z}_q$. A pseudo-semi-functional key update is $\widehat{\mathsf{KU}}_\mathsf{T} := (\{\widehat{\mathsf{KU}}_{\mathsf{T},\theta}\}_{\theta\in\mathsf{KUNode}(\mathsf{BT},\mathsf{RL}_\mathsf{T})}, \mathcal{F}_\mathsf{T})$.

We additionally define the following games.

Game $3_{j,1}$ $(0 \le j \le |\mathcal{T}|-1)$**:** This game is the same as **Game** 3_j except that for a $(j+1)$-st revocation and key update query, a pseudo-normal key update $\overline{\mathsf{KU}}_{j+1}$ is returned.

Game $3_{j,2}$ $(0 \le j \le |\mathcal{T}|-1)$**:** This game is the same as **Game** $3_{j,1}$ except that for a $(j+1)$-st revocation and key update query, a pseudo-semi-functional key update $\widehat{\mathsf{KU}}_{j+1}$ is returned.

We show that for every $j \in \{0,1,\ldots,|\mathcal{T}|-1\}$,

$$\left|\Pr[E_{3_j}] - \Pr[E_{3_{j+1}}]\right| \le \left|\Pr[E_{3_j}] - \Pr[E_{3_{j,1}}]\right| + \left|\Pr[E_{3_{j,1}}] - \Pr[E_{3_{j,2}}]\right| + \left|\Pr[E_{3_{j,2}}] - \Pr[E_{3_{j+1}}]\right|,$$

is negligible.

Lemma 15. *For every $j \in \{0,1,\ldots,|\mathcal{T}|-1\}$, if there exists a PPT adversary $\mathcal{A}$ to distinguish* **Game** 3_j *and* **Game** $3_{j,1}$*, then there exists a PPT adversary $\mathcal{B}$ to break the DDH2 assumption.*

Lemma 16. *For every $j \in \{0, 1, \ldots, |\mathcal{T}| - 1\}$, all of the distributions in* **Game** $3_{j,1}$ *are identical to those in* **Game** $3_{j,2}$ *from the view point of any PPT adversary $\mathcal{A}$. Namely, it holds* $\Pr[E_{3_{j,1}}] = \Pr[E_{3_{j,2}}]$ *for every* $j \in \{0, 1, \ldots, |\mathcal{T}| - 1\}$.

Lemma 17. *For every $j \in \{0, 1, \ldots, |\mathcal{T}| - 1\}$, if there exists a PPT adversary $\mathcal{A}$ to distinguish* **Game** $3_{j,2}$ *and* **Game** 3_{j+1}*, then there exists a PPT adversary $\mathcal{B}$ to break the DDH2 assumption.*

We omit the proof of the above lemmas since those can be proved in the same ways as Lemmas 12, 13, and 14.