Hindawi
Security and Communication Networks
Volume 2020, Article ID 8834872, 16 pages
https://doi.org/10.1155/2020/8834872

Research Article

AKC-Based Revocable ABE Schemes from LWE Assumption

Leixiao Cheng,[1] **Fei Meng**,[2,3] **Xianmeng Meng**,[4] **and Qixin Zhang**[5]

[1]*School of Mathematical Sciences, Fudan University, Shanghai 200433, China*
[2]*School of Mathematics, Shandong University, Jinan, Shandong 250100, China*
[3]*Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Jinan, Shandong 250100, China*
[4]*School of Mathematics, Shandong University of Finance and Economics, Jinan, Shandong 250014, China*
[5]*Tandon School of Engineering, New York University, New York City, USA*

Correspondence should be addressed to Fei Meng; mengfei_sdu@163.com

Received 12 June 2020; Revised 26 August 2020; Accepted 23 September 2020; Published 17 November 2020

Academic Editor: Barbara Masucci

The emergence of quantum computing threatens many classical cryptographic schemes, leading to the innovations in public-key cryptography for postquantum cryptography primitives and protocols that resist to quantum attacks. Lattice-based cryptography is considered to be one of the promising mathematical approaches to achieving security resistant to quantum attacks, which could be built on the learning with errors (LWE) problem and its variants. The fundamental building blocks of protocols for public-key encryption (PKE) and key encapsulation mechanism (KEM) submitted to the National Institute of Standards and Technology (NIST) based on LWE and its variants are called key consensus (KC) and asymmetric key consensus (AKC) by Jin et al. They are powerful tools for constructing PKE schemes. In this work, we further demonstrate the power of KC/AKC by proposing two special types of PKE schemes, namely, revocable attribute-based encryption (RABE). To be specific, on the basis of AKC and PKE/KEM protocols submitted to the NIST based on LWE and its variants, combined with full-rank difference, trapdoor on lattices, sampling algorithms, leftover hash lemma, and binary tree structure, we propose two directly revocable ciphertext-policy attribute-based encryption (DR-ABE) schemes from LWE, which support flexible threshold access policies on multivalued attributes, achieving user-level and attribute-level user revocation, respectively. Specifically, the construction of the ciphertext is derived from AKC, and the revocation list is defined and embedded into the ciphertext by the message sender to revoke a user in the user-level revocable scheme or revoke some attributes of a certain user in the attribute-level revocable scheme. We also discuss how to outsource decryption and reduce the workload for the end user. Our schemes proved to be secure in the standard model, assuming the hardness of the LWE problem. The two schemes imply the versatility of KC/AKC.

1. Introduction

In the 1990s, Shor [1] proposed a quantum algorithm that is capable of solving the integer factorization problem (IFP) and the discrete logarithm problem (DLP) in polynomial time, which aroused the attention of all parties to the development of quantum computers. Practical quantum computing, when available to cyber adversaries, will break the security of nearly all modern public-key cryptographic systems (including RSA and ECC) [2, 3]. In response to the upcoming quantum computer, cryptography researchers have begun to devote themselves to work on replacing the classical public-key cryptosystem with a cryptographic system that can resist quantum attacks, that is, postquantum cryptography [4]. Among all computational problems that believed to be quantum-safe, lattice-based problems emerged as more economical and quantum-safe encryption providers due to their strong security proof, simplicity, and efficient implementation. In particular, the learning with errors (LWE) problem [5] has turned out to be an amazing versatile basis for cryptographic constructions due to its rigorous reduction from the worst case of the lattice problems. Recently, based on the hardness assumptions of the LWE problem and its variants, many postquantum cryptographic schemes [5–17] have been proposed, and they mainly focus on public-key encryption (PKE).

The National Institute of Standards and Technology (NIST) [18] announced a formal call for proposals for postquantum cryptography, which promoted the update of public-key cryptographic algorithms and the research of postquantum cryptographic algorithms. Thereafter, it provided the first-round submissions of postquantum cryptographic standard protocols [19]. Among them, PKE and key encapsulation mechanism (KEM) based on LWE and its variants constitute the dominating set of PKE/KEM proposals. The fundamental building tools of these proposals, i.e., the algorithms that show how to agree on an exact shared key from two close key exchange values, are referred to as key consensus (KC) or asymmetric key consensus (AKC) in [9–11, 16]. The inequation of parameters for any KC and AKC reveals the inherent constraints among security, bandwidth, correctness, and consensus range. KC/AKC and its inequation are the basis for many lattice-based public-key encryption schemes, and they are also powerful tools for constructing public-key encryption.

In this work, we further demonstrate the power of KC/AKC by proposing two special types of public-key encryption schemes, i.e., revocable attribute-based encryption (RABE). As an extension of attribute-based encryption (ABE) [20], RABE [21–26] provides both fine-grained access control on encrypted data and revocation mechanisms when user's attributes change, key exposure, and so on. The revocation mechanism in ABE can be roughly divided into two types: user-level user revocation [27–29] and attribute-level user revocation [30]. In user-level user revocation, when a user leaves the system, he/she should be revoked and cannot decrypt any ciphertext. In attribute-level user revocation, when some attributes of a user are removed, he/she will lose the authorities corresponding to these attributes. The methods for revocation can be divided into two types: indirect revocation [24, 31, 32] and direct revocation [33–35]. In indirect revocation schemes, the authority needs to master the revocation list and issues key update for nonrevoked users regularly. In addition, all nonrevoked users need to communicate with the authority and update their decryption keys periodically as well. However, in direct revocation schemes, the revocation list is defined by the message sender, who "embeds" it into the ciphertext during encryption. Therefore, the authority does not need to generate and issue key update. We find that KC and AKC are fundamental and powerful tools for constructing RABE schemes, combined with full-rank difference (FRD) [36, 37], trapdoor for lattices [38–40], sampling algorithms [36, 41], leftover hash lemma [36], and binary tree structure [31, 42–45].

1.1. Motivation. The basic building blocks of the PKE/KEM protocols based on LWE and its variants submitted to the NIST, namely, KC/AKC, are significant for constructing general or special PKE schemes. Revocable ABE is an advanced form of PKE. Note that the existing lattice-based revocable ABE schemes are limited: [46] cannot resist collusion attacks, while [47] cannot really use the binary tree structure; In addition, as shown in Table 1, their models are incomplete to capture the security requirements for revocable ABE. Therefore, we manage to put forward LWE-based RABE scheme resistant to collusion attacks with a reasonable security model, inspired by PKE/KEM protocols submitted to NIST [19], AKC [10–12, 16], and Zhang et al. [48].

1.2. Our Contributions. In this work, we further demonstrate the power of KC/AKC by proposing two special types of PKE schemes, namely, RABE. To be specific, on the basis of AKC and PKE/KEM protocols submitted to the NIST based on LWE and its variants, combined with full-rank difference, trapdoor on lattices, sampling algorithms, leftover hash lemma, and binary tree structure, we propose two directly revocable ciphertext-policy attribute-based encryption (DR-ABE) schemes from LWE. One achieves user-level user revocation, while the other achieves attribute-level user revocation. Both schemes support flexible threshold access policies on multivalued attributes. The size of the public key of our schemes can be reduced in the random oracle model. The two schemes imply the versatility of KC/AKC. The main advantages of our DR-ABE schemes are as follows:

Multibit encryption: the message sender is allowed to encrypt $M \in \mathbb{Z}_k$ instead of $M \in \{0, 1\}$.

Direct revocation: the revocation list is embedded into the ciphertext by the message sender; the authority does not have to generate and issue key update; all nonrevoked users do not need to communicate with the authority to update their decryption keys.

User-level and attribute-level user revocation: we provide two DR-ABE schemes with user-level and attribute-level user revocation, respectively. We use different techniques to construct these two schemes because the method of constructing the user-level scheme cannot be directly extended to the attribute-level scheme.

Fine-grained access control: our schemes support flexible threshold access policies on multivalued attributes.

Collusion resistance: users in the system cannot combine their information together to illegitimately gain unauthorized data through collaboration.

Resistant against quantum attacks: the security of our schemes is reduced to the learning with errors (LWE) problem.

Decryption outsourced: most computational overhead of the end user in our DR-ABE schemes can be outsourced to a third party (Appendix D).

In Table 1, we compare the features of our schemes with other lattice-based ABE and revocable ABE schemes.

Note that Zhang et al. [48] did not consider revocation. Wang et al. [46] and Kang et al. [47] achieved attribute-level user revocation. In the security model of Wang et al. [46], after submitting the challenge access structure $\mathring{A}A^*$ and challenge revocation list $RL^* = \{RL_i^*\}$, the adversary can only issue key generation queries $(id, S = \{att_i\}_{i \in I})$ under the restriction $S \not\models \mathring{A}A^*$, while in [47], there is a stricter

Table 1: Feature comparison with other schemes.

	Multibit	Direct/indirect	Collusion resistance	Security model	Dec outsourced
Zhang et al. [48]	No	—	Yes	Reasonable	No
Wang et al. [46]	No	Indirect	No	Unreasonable	No
Kang et al. [47]	No	Indirect	Yes	Unreasonable	No
Ours 1	Yes	Direct	Yes	Reasonable	Yes
Ours 2	Yes	Direct	Yes	Reasonable	Yes

restriction, $\text{att}_i \notin \text{ÅA}^*$. However, these restrictions are unreasonable. Because the private key of the key generation query (id, S) should be given to the adversary as long as the nonrevoked attribute set $S_{\text{id,RL}^*} = \{\text{att}_i \in S | \text{id} \notin \text{RL}_i^*, i \in I\}$ does not satisfy ÅA^*, which is the case in our security model for DR-ABE with attribute-level user revocation. In other words, Wang et al. [46] and Kang et al. [47] did not take into account all the key queries that an adversary could issue, while both of our schemes have considered all the situations of the key generation queries from the adversary. In Appendix D, we discuss how to outsource most computational overhead of the end user to an honest-but-curious third party.

In Table 2, we compare the efficiency of our schemes with other lattice-based ABE and revocable ABE schemes. Here, N and M stand for the number of users and attributes in the system, respectively. r_i means the number of revoked users in the i-th attribute binary tree. r_i' represents the number of revoked attributes in the i-th user binary tree. δ is a number such that $n^{1+\delta} > \lceil (n+1)\log q + \omega(\log n) \rceil$.

Zhang et al.'s scheme [48] has relatively small size in every aspect because it does not take revocation into account. Wang et al.'s scheme [46] has smaller public key size than our schemes since it is an indirect revocation mechanism which gives rise to a large updated key size. Kang et al.'s scheme [47] is also an indirect revocation mechanism and has the smallest public key size; however, its security model is relatively unreasonable. It can be seen that the size of the public key and ciphertext in our schemes is larger than that of other schemes. This is because we adopt the direct revocation method, which allows senders to define the revocation list and greatly reduces the workload of the authority. Specifically, the authority does not need to generate and issue updated key periodically. In Appendix C, we describe how to reduce the size of the public key in our schemes in the random oracle model. Briefly speaking, the size of the public key in "our 1" and "our 2" schemes can be reduced from $(2N + M) \cdot \widetilde{o}(n^{2+\delta})$ and $2NM \cdot \widetilde{o}(n^{2+\delta})$ to $M \cdot \widetilde{o}(n^{2+\delta})$ and $M \cdot \widetilde{o}(n^{2+\delta})$, respectively.

1.3. Related Works. The underlying consensus mechanism of most PKE/KEM protocols submitted to the NIST based on LWE and its variants is based on KC/AKC and its variants [16, 49–60]. Specifically, [16, 51, 53] are based on the learning with rounding (LWR) problem [61] and its variant. [50] is based on both the LWE and the LWR problems. [16, 49, 52, 54, 56, 60] are based on the LWE or the ring learning with errors [62] or the module learning with errors [63] problems. To further reduce the error probability, the underlying consensus mechanism for some of PKE/KEM protocols submitted to the NIST based on LWE and its variants additionally employs some error correction codes [16, 54, 64, 65], while others directly use lattice codes [16, 66].

Attribute-based encryption (ABE) [20] is a promising cryptographic primitive of public-key encryption that provides fine-grained access control on encrypted data. In 2006, Goyal et al. [67] extended the idea of ABE and classified ABE as key-policy ABE (KP-ABE) [68, 69] and ciphertext-policy ABE (CP-ABE) [70, 72]. In a KP-ABE scheme, the private key of a user is associated with an access policy, while the ciphertext is associated with a set of attributes. On the contrary, in a CP-ABE scheme, the private key of a user is associated with a set of attributes, and the ciphertext is associated with an access policy. Generally, CP-ABE is more flexible than KP-ABE since the former allows users to set their access policies when encrypting messages.

Many revocable attribute-based encryption schemes [21–26] based on classic assumptions (e.g., pairing-related assumptions) have been proposed. However, these schemes would not be secure against attacks from quantum computes. To mitigate this issue, Wang et al. [46] and Kang et al. [47] proposed indirectly revocable CP-ABE schemes from lattices. Both of their schemes had achieved attribute-level user revocation. However, Wang et al. [46] did not resist to collusion attacks, that is, two users who do not satisfy the access structure can successfully decrypt the ciphertext through cooperation. In Kang et al. [47], they built N user binary trees $\{\text{BT}_i\}_{i \in [1,N]}$, where N is the maximum number of users. Each binary tree has M leaf nodes, and each attribute is assigned to a leaf node in the binary tree, where M is the number of attributes in the system. To revoke r' attributes of a user, the authority actually needs to issue $M - r'$-(rather than $r'\log(M/r')$ as they claimed) associated key update in the key updating phase since each attribute is assigned a different secret-shared key. In other words, they did not actually take advantage of the binary-tree data structure to reduce the burden of the authority during the key updating phase as [24, 31, 32].

2. Preliminaries

For notational convenience, we sometimes regard a matrix as simply a set of its column vectors. For a matrix $\mathbf{T}$, let $\|\mathbf{T}\|$ denote the L_2 length of its longest column, i.e., $\|\mathbf{T}\| \coloneqq \max_i \|\mathbf{t}_i\|$; let $s_1(\mathbf{T})$ denote the largest singular value of $\mathbf{T}$, i.e., $s_1(\mathbf{T}) \coloneqq \sup_{\|u\|=1} \|\mathbf{T}\mathbf{u}\|$. Furthermore, if the columns of $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_k\}$ are linearly independent, let $\widetilde{\mathbf{T}} \coloneqq \{\widetilde{\mathbf{t}}_1, \ldots, \widetilde{\mathbf{t}}_k\}$ denote the Gram–Schmidt orthogonalization of vectors

Table 2: Efficiency comparison with other schemes.

	pk size	sk size	Updated key size	Ciphertext size
Zhang et al. [48]	$M \cdot \widetilde{o}(n^{2+\delta})$	$M \cdot \widetilde{o}(n^{1+\delta})$	—	$M \cdot \widetilde{o}(n^{1+\delta})$
Wang et al. [46]	$M \cdot \widetilde{o}(n^{2+\delta})$	$M \cdot O(\log N) \cdot \widetilde{o}(n^{1+\delta})$	$\sum_{i=1}^{M} r_i \log(N/r_i) \cdot \widetilde{o}(n^{1+\delta})$	$M \cdot \widetilde{o}(n^{1+\delta})$
Kang et al. [47]	$\widetilde{o}(n^{2+\delta})$	$M \cdot O(\log M) \cdot \widetilde{o}(n^{1+\delta})$	$\sum_{i=1}^{N} (M - r_i') \cdot \widetilde{o}(n^{1+\delta})$	$M \cdot \widetilde{o}(n^{1+\delta})$
Ours 1	$(2N + M) \cdot \widetilde{o}(n^{2+\delta})$	$M \cdot O(\log N) \cdot \widetilde{o}(n^{1+\delta})$	—	$(r \log(N/r) + M) \cdot \widetilde{o}(n^{1+\delta})$
Ours 2	$2NM \cdot \widetilde{o}(n^{2+\delta})$	$M \cdot O(\log N) \cdot \widetilde{o}(n^{1+\delta})$	—	$(\sum_{i=1}^{M} r_i \log(N/r_i) + M) \cdot \widetilde{o}(n^{1+\delta})$

$\mathbf{t}_1, \ldots, \mathbf{t}_k$ taken in that order. For two matrices $\mathbf{X} \in \mathbb{R}^{n \times m_1}$ and $\mathbf{Y} \in \mathbb{R}^{n \times m_2}$, let $(\mathbf{X} \| \mathbf{Y}) \in \mathbb{R}^{n \times (m_1 + m_2)}$ denote the concatenation of the columns of $\mathbf{X}$ followed by the columns of $\mathbf{Y}$. For two matrices $\mathbf{X} \in \mathbb{R}^{n_1 \times m}$ and $\mathbf{Y} \in \mathbb{R}^{n_2 \times m}$, let $(\mathbf{X}; \mathbf{Y}) \in \mathbb{R}^{(n_1 + n_2) \times m}$ denote the concatenation of the rows of $\mathbf{X}$ followed by the rows of $\mathbf{Y}$.

For nonnegative integers $i < j$, let $[i, j]$ denote the set $\{i, i+1, \ldots, j\}$. If S is an attribute set and ÅA is an access structure, then $S \vDash$ ÅA means that S satisfies ÅA. If S is a finite set, then $x \longleftarrow S$ is the operation of choosing an element uniformly at random from S. For a probability distribution $\mathscr{D}$, $x \longleftarrow \mathscr{D}$ denotes the operation of choosing an element according to $\mathscr{D}$. If γ is either an algorithm or a set, then $x \longleftarrow \gamma$ is a simple assignment statement.

The natural security parameter throughout this paper is n. A function $f(n)$ is negligible, denoted as $\mathrm{negl}(n)$, if for every $c > 0$, there exists n_c such that $f(n) < 1/n^c$ for all $n > n_c$. We say that a probability is overwhelming if it is $1 - \mathrm{negl}(n)$. An algorithm is probabilistic polynomial-time (PPT) computable if it is modeled as a probabilistic Turing machine whose running time is bounded by some polynomial function.

2.1. Directly Revocable Attribute-Based Encryption. A directly revocable ciphertext-policy attribute-based encryption (DR-ABE) scheme with user-level (resp. attribute-level) user revocation consists of the following four algorithms {Setup, Keygen, Enc, Dec}:

Setup $(n, \mathscr{R}, N)$: this algorithm takes as input a security parameter n, a system attribute set $\mathscr{R}$, and a maximal number of users N in the system and returns a public key PK and a master secret key MSK.

Keygen (PK, MSK, id, S): this algorithm takes as input a public key PK, a master secret key MSK, an identity id, and an attribute set $S = \{\mathrm{att}_i\} \subseteq \mathscr{R}$ for the user with identity id and returns a private key $\mathrm{sk}_{S,\mathrm{id}}$.

Enc (PK, ÅA, RL, M): this algorithm takes as input a public key PK, an access structure ÅA = $(\mathrm{W} = \{\mathrm{att}_j\}_{j \in J}, t)$, a revocation list RL (resp. a family of attribute revocation lists $\mathrm{RL} = \{\mathrm{RL}_j\}_{j \in J}$, where RL_j consist of identities whose j-th attribute is revoked), and a message M and returns a ciphertext C.

Dec (PK, $\mathrm{sk}_{S,\mathrm{id}}$, C): this algorithm takes as input a public key PK, a private key $\mathrm{sk}_{S,\mathrm{id}}$ of identity id with attribute set $S = \{\mathrm{att}_i\}$, and a ciphertext C encrypted under access structure ÅA and RL; it first checks whether $S \vDash$ ÅA and id $\notin$ RL (resp. whether the set of nonrevoked attributes of the identity id, $S_{\mathrm{id,RL}} = \{\mathrm{att}_i \in S | \mathrm{id} \notin \mathrm{RL}_i\} \vDash$ ÅA). If not, the algorithm returns a special symbol $\perp$ indicating decryption failure. Otherwise, it returns a message M.

Note that, for the DR-ABE scheme with attribute-level user revocation, it is reasonable that the message sender only needs to consider attribute revocation lists associated with his/her access structure.

2.2. Security Model for DR-ABE. We now describe the selective security model for the DR-ABE scheme with user-level (resp. attribute-level) user revocation. The security model is described by the following game between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$.

Init: the adversary $\mathscr{A}$ chooses an access structure $A^* = (W^*, t^*)$ with $W^* = \{\mathrm{att}_j^*\}_{j \in J^*}$ and a revocation list RL^* (resp. a family of attribute revocation lists $\mathrm{RL}^* = \{\mathrm{RL}_j^*\}_{j \in J^*}$) and submits them to the challenger $\mathscr{C}$.

Setup: $\mathscr{C}$ runs the Setup algorithm, gives the public key PK to $\mathscr{A}$, and keeps the master secret key MSK private.

Phase 1: $\mathscr{A}$ can adaptively make a number of key generation queries (id, S), where $S = \{\mathrm{att}_i\}_{i \in I}$. The restriction is that if $S \vDash W^*$, then id $\in \mathrm{RL}^*$ (resp. the nonrevoked attribute set $S_{\mathrm{id,RL}^*} = \{\mathrm{att}_i \in S | \mathrm{id} \notin \mathrm{RL}_i^*, i \in I\}$ does not satisfy A^*).

Challenge: $\mathscr{A}$ submits two equal-length messages, $M_0 \neq M_1$. The challenger $\mathscr{C}$ flips a random coin $b \in \{0, 1\}$, computes $C^* = \mathrm{Enc}(\mathrm{PK}, A^*, \mathrm{RL}^*, M_b)$, and gives C^* to $\mathscr{A}$.

Phase 2: it is the same as in Phase 1.

Guess: $\mathscr{A}$ output a guess $b' \in \{0, 1\}$ for b.

The advantage of adversary $\mathscr{A}$ in the above game is defined as

$$\mathrm{Adv}_{\mathscr{A}}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right|. \tag{1}$$

Definition 1. A directly revocable ciphertext-policy attribute-based encryption scheme is secure if the advantage $\mathrm{Adv}_{\mathscr{A}}(\lambda)$ is negligible in λ for all polynomial-time adversaries $\mathscr{A}$.

2.3. Background on Lattices. Let $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_m\} \subset \mathbb{R}^m$ consist of m linearly independent vectors. The m-dimensional full-rank lattice Λ generated by the basis $\mathbf{B}$ is the set $\Lambda = \mathscr{L}(\mathbf{B}) := \{\sum_{i=1}^{m} x_i \mathbf{b}_i | x_i \in \mathbb{Z}\}$. For any positive integers

n, m, and $q \geq 2$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, we define $\mathscr{L}_q^{\perp}(\mathbf{A}) \coloneqq \{\mathbf{z} \in \mathbb{Z}^m\colon \mathbf{A}\cdot\mathbf{z} = \mathbf{0}_n \bmod q\}$ and $\mathscr{L}_q^{u}(\mathbf{A}) \coloneqq \{\mathbf{z} \in \mathbb{Z}^m\colon \mathbf{A}\cdot\mathbf{z} = \mathbf{u} \bmod q\}$.

2.3.1. Discrete Gaussian. Let Λ be an m-dimensional lattice. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any parameter $\sigma \in \mathbb{R}_{>0}$, define $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi(\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2))$ and $\rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{x\in\Lambda}\rho_{\sigma,\mathbf{c}}(\mathbf{x})$. The discrete Gaussian distribution over Λ with center $\mathbf{c}$ and Gaussian parameter σ is $\mathscr{D}_{\Lambda,\sigma,\mathbf{c}} = (\rho_{\sigma,\mathbf{c}(y)}/\rho_{\sigma,\mathbf{c}}(\Lambda))$ for $\forall \mathbf{y} \in \Lambda$. If $\mathbf{c} = \mathbf{0}$, we conveniently use ρ_σ and $\mathscr{D}_\Lambda\sigma$. In the following, we summarize some basic properties of the discrete Gaussian distribution.

Lemma 1 (see [73]). *Let n, m, and q be positive integers with $m > n$, $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix, $\mathbf{u} \in \mathbb{Z}_q^n$ be a vector, $\mathbf{T}_A$ be a basis for $\Lambda = \mathscr{L}_q^u(\mathbf{A})$, and $\sigma \geq \|\widetilde{\mathbf{T}_A}\| \cdot \omega(\sqrt{\log m})$. Then, $Pr[\|\mathbf{x}\| > \sigma\sqrt{m}\colon \mathbf{x} \longleftarrow \mathscr{D}_\Lambda\sigma] \leq negl(n)$.*

Lemma 2 (see [73]). *Let $n, m, q > 0$ be positive integers with $m \geq 2n\lceil\log q\rceil$ and q being a prime. Let σ be any positive real number such that $\sigma \geq \omega(\sqrt{\log m})$. Then, for $\mathbf{A} \longleftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{e} \longleftarrow \mathscr{D}_{\mathbb{Z}^m}\sigma$, the distribution of $\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. Furthermore, for fixed $\mathbf{u} \in \mathbb{Z}_q^n$, the conditional distribution of $\mathbf{e} \longleftarrow \mathscr{D}_{\mathbb{Z}^m}\sigma$, given $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ for uniformly random $\mathbf{A}$ in $\mathbb{Z}_q^{n\times m}$, is $\mathscr{D}_{\Lambda_q^u(A)}\sigma$ with all but negligible probabilities.*

2.4. The LWE Hardness Assumption. Security of our construction reduces to the learning with errors (LWE) problem defined by Regev [5].

Definition 2. Consider a prime q, a positive integer n, and a distribution χ over $\mathbb{Z}_q$, all public. A $(\mathbb{Z}_q, n, \chi)$-LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being either a noisy pseudo-random sampler $\mathcal{O}_s$ carrying some constant random secret key $\mathbf{s} \in \mathbb{Z}_q^n$ or a truly random sampler $\mathcal{O}_\$$, whose behaviors are, respectively, as follows:

> $\mathcal{O}_s$**:** outputs samples in $\mathbb{Z}_q^n \times \mathbb{Z}_q$ of the form $(\mathbf{u}_i, v_i) = (\mathbf{u}_i, \mathbf{u}_i^\top\mathbf{s} + x_i)$, where $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniformly distributed persistent value invariant across invocations, $x_i \in \mathbb{Z}_q$ is a fresh sample from χ, and $\mathbf{u}_i$ is uniform in $\mathbb{Z}_q^n$
>
> $\mathcal{O}_\$$**:** outputs truly uniform random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$

The $(\mathbb{Z}_q, n, \chi)$-LWE problem allows repeated queries to the challenge $\mathcal{O}$. We say that an algorithm $\mathscr{A}$ decides the $(\mathbb{Z}_q, n, \chi)$-LWE problem if $|\Pr[\mathscr{A}^{\mathcal{O}_s} = 1] - \Pr[\mathscr{A}^{\mathcal{O}_\$} = 1]|$ is nonnegligible for random $\mathbf{s} \in \mathbb{Z}_q^n$.

Regev [5] and Peikert [74] showed that, for certain noise distribution χ, denoted as $\overline{\Psi}_\alpha$, the LWE problem is hard.

Definition 3. Consider a real number $\alpha = \alpha(n) \in (0,1)$ and a prime q. Let $\mathbb{T} \coloneqq \mathbb{R}/\mathbb{Z}$ be the group of real numbers $[0,1)$ with addition modulo 1. Define by Ψ_α the distribution over $\mathbb{T}$ of a normal variable with mean 0, standard deviation $\alpha/\sqrt{2\pi}$, and reduced modulo 1, i.e.,

$$\Psi_\alpha(r) \coloneqq \sum_{k=-\infty}^{\infty} \frac{1}{\alpha}\cdot \exp\left(-\pi\left(\frac{r-k}{\alpha}\right)^2\right), \quad \forall r \in [0,1). \tag{2}$$

We denote by $\overline{\Psi}_\alpha$ the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q \cdot X_{\Psi_\alpha}\rfloor \mod q$, where the random variable $X_{\Psi_\alpha} \in \mathbb{T}$ has distribution Ψ_α.

Lemma 3. *Consider $\alpha = \alpha(n) \in (0,1)$ and a prime $q = q(n)$ such that $\alpha q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm which solves the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem, then there exists an efficient quantum algorithm for approximating SIVP in the ℓ_2 norm, in the worst case, to within $\widetilde{O}(n/\alpha)$ factors.*

The following lemma about the distribution $\overline{\Psi}_\alpha$ will be used to analyze the correctness of our constructions in Sections 4 and 5.

Lemma 4 (see [36]). *Let $\mathbf{e}$ be some vector in $\mathbb{Z}^m$ and $\mathbf{x} \longleftarrow \overline{\Psi}_\alpha^m$. Then, the quantity $|\mathbf{e}^\top\mathbf{x}|$, treated as an integer in $[0, q-1]$, satisfies*

$$\left|\mathbf{e}^\top\mathbf{x}\right| \leq \|\mathbf{e}\| q\alpha\omega(\sqrt{\log m}) + \|\mathbf{e}\|\frac{\sqrt{m}}{2}, \tag{3}$$

with all but negligible probabilities in m. In particular, if $x \longleftarrow \overline{\Psi}_\alpha$ is treated as an integer in $[0, q-1]$, then $|x| \leq q\alpha\omega(\sqrt{\log m}) + 1/2$ with all but negligible probabilities in m.

3. Technical Tools

In this section, we introduce the notion of AKC given in [9, 11, 17] and some other related technical tools in this paper.

3.1. Asymmetric Key Consensus

Definition 4. An asymmetric key consensus scheme AKC = (params, Con, Rec) is specified as follows:

(i) params $= (q, k, g, \overline{d}, \text{aux})$ denotes the system parameters, where $q, 2 \leq k, g \leq q, 1 \leq \overline{d} \leq \lfloor q/2\rfloor$ are positive integers and aux denotes some auxiliary values that are usually determined by $(q, k, g, \overline{d})$ and could be set to be empty.

(ii) $v \longleftarrow \text{Con}(\sigma_1, k_1, \text{params})$: on inputting ($\sigma_1 \in \mathbb{Z}_q$, $k_1 \in \mathbb{Z}_k$, params), the probabilistic polynomial-time conciliation algorithm Con outputs the public hint signal $v \in \mathbb{Z}_g$.

(iii) $k_2 \longleftarrow \text{Rec}(\sigma_2, v, \text{params})$: on inputting ($\sigma_2, v$, params), the deterministic polynomial-time algorithm Rec outputs $k_2 \in \mathbb{Z}_k$.

> Correctness: an AKC scheme is correct if it holds $k_1 = k_2$ for any $\sigma_1, \sigma_2 \in \mathbb{Z}_q$ such that $|\sigma_1 - \sigma_2|_q \leq \overline{d}$.
> Security: an AKC scheme is secure if v is independent of k_1 whenever σ_1 is uniformly distributed over $\mathbb{Z}_q$. Specifically, for arbitrary $\tilde{v} \in \mathbb{Z}_g$ and

arbitrary $\widetilde{k}_1, \widetilde{k}_1' \in \mathbb{Z}_k$, it holds that $\Pr[v = \widetilde{v} | k_1 = \widetilde{k}_1] = \Pr[v = \widetilde{v} | k_1 = \widetilde{k}_1']$, where the probability is taken over $\sigma_1 \longleftarrow \mathbb{Z}_q$ and the random coins are used by Con.

Theorem 1 (see [11]). *Let AKC be an asymmetric key consensus scheme with parameters params* $= (q, k, \overline{d}, g, aux)$. *If AKC is correct and secure, then* $2k\overline{d} \le q(1 - (k/g))$.

Next, we review the construction and analysis of the instantiated AKC called asymmetric key consensus with noise (AKCN) in [11]. The illustration diagram is given in Algorithm 1. When the parameters $q = 2^{\overline{q}}$ and $k = 2^{\overline{k}}$ are powers of 2, AKCN can be simplified as AKCN power 2 [9].

Theorem 2 (see [11]). *Suppose the parameters of AKCN satisfy* $(2\overline{d} + 1)k < q(1 - (k/g))$, *the AKCN scheme described in Algorithm 1 is correct.*

Theorem 3 (see [11]). *The AKCN scheme is secure, i.e., v is independent of* k_1 *when* $\sigma_1 \longleftarrow \mathbb{Z}_q$.

3.2. Full-Rank Difference Encoding (FRD). In our construction and proof of security, we need an encoding function H: $\mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n \times n}$ to map attributes in $\mathbb{Z}_q^n$ to matrices in $\mathbb{Z}_q^{n \times n}$.

Definition 5 (see [36, 37]). Let q be a prime and n a positive integer. We say that a function H: $\mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n \times n}$ is encoding with full-rank difference (FRD) if

(1) For all distinct $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$, the matrix $H(\mathbf{x}) - H(\mathbf{y})$ is full rank

(2) $\mathbf{G}_{\mathrm{FRD}}$ is computable in polynomial time

3.3. Trapdoors for Lattices. We review two trapdoor generation algorithms in the following lemma. The first algorithm generates a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ that is statistically close to uniform, together with a short trapdoor basis for the associated lattice $\Lambda_q^\perp(\mathbf{A})$. The second algorithm generates a basis for the lattice $\Lambda_q^\perp(\mathbf{G})$, where $\mathbf{G}$ is what they call the primitive matrix.

Lemma 5 (see [38–40]). *Let* $n, m, q > 0$ *be positive integers with* $m \ge 2n\lceil \log q \rceil$ *and q being a prime. Then, we have*

(i) [38–40], a PPT algorithm TrapGen that outputs a pair $(\mathbf{A}, \mathbf{T}_A) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{m \times m}$ *such that* $\mathbf{A}$ *is full rank and statistically close to uniform and* $\mathbf{T}_A$ *is a basis for* $\Lambda_q^\perp(\mathbf{A})$ *satisfying* $\|\widetilde{\mathbf{T}_A}\| \le O(\sqrt{n \log q})$

(ii) [40], 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* $\mathbf{T}_G \in \mathbb{Z}^{m \times m}$ *with* $\|\widetilde{\mathbf{T}_G}\| \le \sqrt{5}$

3.4. Sampling Algorithms. The following SampleLeft [36, 41] and SampleRight [36] algorithms will be used to sample short vectors in our construction and in the simulation, respectively.

Lemma 6. *Let integers* $q > 2$ *and* $m > n$. *There is an efficient PPT algorithm SampleLeft* $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{T}_A, \sigma)$ *which takes as input a full-rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, *a matrix* $\mathbf{B} \in \mathbb{Z}_q^{n \times \overline{m}}$, *a vector* $\mathbf{u} \in \mathbb{Z}_q^n$, *a basis* $\mathbf{T}_A \in \mathbb{Z}^{m \times m}$ *of* $\Lambda_q^\perp(\mathbf{A})$, *and a Gaussian parameter* $\sigma > \|\widetilde{\mathbf{T}_A}\| \cdot \omega(\sqrt{\log(m + \overline{m})})$ *outputs a vector* $\mathbf{e} \in \mathbb{Z}^{m + \overline{m}}$ *distributed statistically close to* $\mathcal{D}_{\Lambda_q^u([A \| B])}\sigma$.

Lemma 7. *Let integers* $q > 2$ *and* $m > n$. *There is an efficient PPT algorithm SampleRight* $(\mathbf{A}, \mathbf{B}, \mathbf{Ru}, \mathbf{T}_B, \sigma)$ *which takes as input matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$, *where* $\mathbf{B}$ *is full rank, a uniform random matrix* $\mathbf{R} \in \mathbb{Z}^{m \times m}$, *a vector* $\mathbf{u} \in \mathbb{Z}_q^n$, *a basis* $\mathbf{T}_B$ *of* $\Lambda_q^\perp(\mathbf{B})$, *and a Gaussian parameter* $\sigma > \|\widetilde{\mathbf{T}_B}\| \cdot s_1(\mathbf{R}) \cdot \omega(\sqrt{\log m})$, *and outputs a vector* $\mathbf{e} \in \mathbb{Z}^{2m}$ *distributed statistically close to* $\mathcal{D}_{\Lambda_q^u([A \| AR + B])}\sigma$.

3.5. Leftover Hash Lemma. To prove correctness and security of our construction, we need more lemmas from [36] as follows.

Lemma 8. *Let* $\mathbf{R}$ *be an* $m \times m$ *matrix chosen at random from* $\{-1, 1\}^{m \times m}$; *then, there exists a universal constant C such that* $Pr[s_1(\mathbf{R}) > C\sqrt{m}] < e^{-m}$.

Lemma 9 (leftover hash lemma). *Suppose thatq is a prime and that* $m > (n + 1)\log q + \omega(\log n)$. *Let* $\mathbf{A}, \mathbf{B}$ *be matrices chosen uniformly in* $\mathbb{Z}_q^{n \times m}$ *and* $\mathbf{R}$ *be an* $m \times m$ *matrix chosen uniformly in* $\{-1, 1\}^{m \times m} \bmod q$. *Then, for all vectors* $\mathbf{w}$ *in* $\mathbb{Z}_q^m$, *the distribution* $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top \mathbf{w})$ *is statistically close to the distribution* $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{w})$.

3.6. The Binary-Tree Data Structure. Our construction makes use of the binary-tree data structure, as with [31, 42–45]. This structure uses a node selection algorithm called KUNodes. In the algorithm, we use the following notations: BT denotes a binary tree. root denotes the root node of BT. θ denotes a node in the binary tree, and v emphasizes that the node θ is a leaf node. The set $\mathrm{Path}(\mathrm{BT}, v)$ stands for the collection of nodes on the path from the leaf v to the root (including v and the root). If θ is a nonleaf node, then θ_ℓ and θ_r denote the left and right child of θ, respectively. The KUNodes algorithm takes as input a binary tree BT and a revocation list RL and outputs a set of nodes Y, which is the smallest subset of nodes that contains an ancestor of all the leaf nodes corresponding to nonrevoked indexes. The description of the KUNodes algorithm is as follows:

KUNodes(BT, RL):

$X, Y \longleftarrow \varnothing$; $\forall v \in \mathrm{RL}$; add $\mathrm{Path}(\mathrm{BT}, v)$ to X
$\forall \theta \in X$: if $\theta_\ell \notin X$, then add θ_ℓ to Y; if $\theta_r \notin X$, then add θ_r to Y
If $Y = \varnothing$, then add *root* to Y; return Y

(1) params = (q, k, g, d, aux), where $aux = \varnothing$.
(2) **procedure con** $(\sigma_1, k_1, \text{params}) \triangleright \sigma_1 \in [0, q-1]$
(3) $\quad v = \lfloor g \cdot (\sigma_1 + \lfloor k_1 q/k \rceil)/q \rceil \bmod g$
(4) $\quad$ **return** v
(5) **end procedure**
(6) **procedure Rec** $(\sigma_2, v, \text{params}) \triangleright \sigma_2 \in [0, q-1]$
(7) $\quad k_2 = \lfloor k \cdot (v/g - \sigma_2/q) \rceil \bmod k$
(8) $\quad$ **return** k_2
(9) **end procedure**

Algorithm 1: AKCN: asymmetric KC with noise.

4. DR-ABE with User-Level User Revocation

KC/AKC is fundamental and powerful for constructing PKE schemes. To demonstrate the versatility of KC/AKC, we propose two DR-ABE schemes from lattices based on AKCN (Algorithm 1), which supports user-level user revocation and attribute-level user revocation in Sections 4 and 5, respectively.

4.1. Construction Details. The main ideas behind our construction can be described as follows. We assign identity id to a leaf node v_{id} in the binary tree BT. Then, we store the attribute set S of id in every node $\theta \in \text{path}(\text{BT}, v_{\text{id}})$: for each θ, the random vector $\mathbf{u}$ in the public key is secret-shared into vectors $\{\hat{\mathbf{u}}_{\theta,i}\}$, where $\hat{\mathbf{u}}_{\theta,i}$ is associated with attribute att_i. If $\text{id} \notin \text{RL}$ and $S \vDash$ ÅA, then there exists a node $\theta^* \in \text{path}(\text{BT}, v_{\text{id}}) \cap \text{KUNodes}(\text{BT}, \text{RL})$, and $\mathbf{u}$ can be recovered using $\{\hat{\mathbf{u}}_{\theta^*,i}\}$.

For convenience, it is assumed that there are ℓ attributes in our system, and the i-th attribute is associated with a value space $\mathcal{R}_i \subseteq \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$. Let $\mathcal{R} = \mathcal{R}_1 \times \cdots \times \mathcal{R}_\ell$ denote the attribute space. We also define d default attributes $\{\ell+1, \ldots, \ell+d\}$. Let $\mathcal{I} = \{1, \ldots, \ell+d\}$ and $\mathcal{I}_1 = \{1, \ldots, \ell\}, \mathcal{I}_2 = \{\ell+1, \ldots, \ell+d\}$, and $D = ((\ell+d)!)^2$.

Setup $(n, \mathcal{R}, N)$: on inputting a security parameter n, a system attribute set $\mathcal{R} = \mathcal{R}_1 \times \cdots \times \mathcal{R}_\ell$, and a maximal number of users N in the system, this algorithm sets the primitive matrix $\mathbf{G}$ (with public trapdoor $\mathbf{T}_G$, see Lemma 5) and the parameters q, m, α, σ, k, g, and $\overline{d}$ as specified in Section 4.4. Then, it performs as follows:

(1) Run $(\mathbf{A}, \mathbf{T}_A) \longleftarrow \text{TrapGen}(n, m, q)$.
(2) Choose $\mathbf{B}_i \longleftarrow \mathbb{Z}_q^{n \times m}$ for $i \in \mathcal{I}$.
(3) Choose $\mathbf{u} \longleftarrow \mathbb{Z}_q^n$.
(4) Choose a full-rank difference map H: $\mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n \times n}$.
(5) Build a binary tree BT with N leaf nodes. For each node $\theta \in$ BT, choose "identifier" $\mathbf{D}_\theta \longleftarrow \mathbb{Z}_q^{n \times m}$.
(6) Return $\text{PK} = \{\mathbf{A}, \{\mathbf{B}_i\}_{i \in \mathcal{I}}, \mathbf{u}, H, \text{BT}\}$ and $\text{MSK} = \mathbf{T}_A$.

Keygen (PK, MSK, id, S): on inputting the public key PK, the master secret key MSK, an identity id, and the attribute set $S = \{\text{att}_i\}_{i \in I}$ of id, where $I \subseteq \mathcal{I}_1$ and $\text{att}_i \in \mathcal{R}_i$, it goes as follows:

(1) Pick an unassigned leaf node v_{id} from BT and store id in that node. For each $\theta \in \text{path}(\text{BT}, v_{\text{id}})$, randomly choose n degree d polynomials $p_{\theta,1}(x), \ldots, p_{\theta,n}(x) \in \mathbb{Z}_q[x]$ such that $\mathbf{u} = (p_{\theta,1}(0), \ldots, p_{\theta,n}(0))^\top$. For each $i \in I \cup \mathcal{I}_2$, let $\hat{\mathbf{u}}_{\theta,i} = (p_{\theta,1}(i), \ldots, p_{\theta,n}(i))^\top$.
(2) For each $\theta \in \text{path}(\text{BT}, v_{\text{id}})$, sample

$$\mathbf{e}_{\theta,i} \longleftarrow \text{SampleLeft}\left(\mathbf{A}, \mathbf{D}_\theta \| \mathbf{B}_i + H(\text{att}_i)\mathbf{G}, \hat{\mathbf{u}}_{\theta,i}, \mathbf{T}_A, \sigma\right), \quad (4)$$

for $i \in I$, and sample $\mathbf{e}_{\theta,i} \longleftarrow SampleLeft(\mathbf{A}, \mathbf{D}_\theta \| \mathbf{B}_i + \mathbf{G}, \hat{\mathbf{u}}_{\theta,i}, \mathbf{T}_\mathbf{A}, \sigma)$, for $i \in \mathcal{I}_2$.

Let $\mathbf{E}_{\theta,i} = (\mathbf{A} \| \mathbf{D}_\theta \| \mathbf{B}_i + H(\text{att}_i)\mathbf{G})$ for $i \in I$ and $\mathbf{E}_{\theta,i} = (\mathbf{A} \| \mathbf{D}_\theta \| \mathbf{B}_i + \mathbf{G})$ for $i \in \mathcal{I}_2$; note that $\mathbf{E}_{\theta,i} \cdot \mathbf{e}_{\theta,i} = \hat{\mathbf{u}}_{\theta,i}$.
(3) Return $\text{sk}_{S,\text{id}} = (\{\mathbf{e}_{\theta,i}\}_{\theta \in \text{path}(\text{BT}, v_{\text{id}}), i \in I \cup \mathcal{I}_2})$ as the private key.

Note that for any $\theta \in \text{path}(\text{BT}, v_{\text{id}})$ and any subset $K \subseteq I \cup \mathcal{I}_2$ with $|K| = d+1$, we have $\mathbf{u} = \sum_{i \in K} L_i \cdot \hat{\mathbf{u}}_{\theta,i}$, where the Lagrange coefficient $L_i = (\prod_{j \in K, j \neq i}(-j)/\prod_{j \in K, j \neq i}(i-j))$.

Enc (PK, (W, t), RL, M): on inputting a public key PK, an attribute set $W = \{\text{att}_j\}_{j \in J_1}$, an integer $1 \le t \le \min(|W|, d)$, a revocation list RL consisting of revoked identities, and a message $M \in \mathbb{Z}_k$, it works as follows:

(1) Choose $\mathbf{s} \longleftarrow \mathbb{Z}_q^n$ and compute

$$c_0 = \text{Con}(\mathbf{u}^\top \mathbf{s} + Dx_0, M) = \lfloor g \cdot \frac{(\mathbf{u}^\top \mathbf{s} + Dx_0 + \lfloor Mq/k \rceil)}{q} \rceil \bmod g, \quad (5)$$

and $\mathbf{c} = \mathbf{A}^\top \mathbf{s} + D\mathbf{x}$, where $x_0 \longleftarrow \overline{\Psi}_\alpha$ and $\mathbf{x} \longleftarrow \overline{\Psi}_\alpha^m$.
(2) For each $j \in J_1$, choose $\mathbf{R}_j \longleftarrow \{-1, 1\}^{m \times m}$ and compute $\mathbf{c}_j = (\mathbf{B}_j + H(\text{att}_j)\mathbf{G})^\top \mathbf{s} + D\mathbf{R}_j^\top \mathbf{x}$.
(3) Let $J_2 = \{\ell+1, \ldots, \ell+d+1-t\}$, and for each $j \in J_2$, choose $\mathbf{R}_j \longleftarrow \{-1, 1\}^{m \times m}$ and compute $\mathbf{c}_j = (\mathbf{B}_j + \mathbf{G})^\top \mathbf{s} + D\mathbf{R}_j^\top \mathbf{x}$.
(4) For each $\theta \in \text{KUNodesBTRL}$, choose $\mathbf{R}_\theta \longleftarrow \{-1, 1\}^{m \times m}$ and compute $\mathbf{c}_\theta = \mathbf{D}_\theta^\top \cdot \mathbf{s} + D\mathbf{R}_\theta^\top \cdot \mathbf{x}$.
(5) Return $C = (c_0, \mathbf{c}, \{\mathbf{c}_j\}_{j \in J_1 \cup J_2}, \{\mathbf{c}_\theta\}_{\theta \in \text{KUNodesBTRL}})$ as the ciphertext.

$\mathrm{Dec}(\mathrm{PK}, \mathrm{sk}_{S,\mathrm{id}}, C)$: on inputting the public key PK, the private key $\mathrm{sk}_{S,\mathrm{id}}$ of identity id with attribute set $S = \{\mathrm{att}_i\}_{i\in I}$, and a ciphertext C encrypted under access structure $(W = \{\mathrm{att}_j\}_{j\in J_1}, t)$ and revocation list RL,

(1) If $|S \cap W| < t$ or $\mathrm{id} \in \mathrm{RL}$, return $\perp$.

(2) Else, parse the private key $\mathrm{sk}_{S,\mathrm{id}} = (\{\mathbf{e}_{\theta,i}\}_{\theta\in\mathrm{path}(\mathrm{BT},\nu_{\mathrm{id}}), i\in I\cup\mathcal{F}_2}$ and $C = (c_0, \mathbf{c}, \{\mathbf{c}_j\}_{j\in J_1\cup J_2}, \{\mathbf{c}_\theta\}_{\theta\in\mathrm{KUNodesBTRL}})$. Since $\mathrm{id} \notin \mathrm{RL}$, there exists $\theta \in \mathrm{path}(\mathrm{BT}, \nu_{\mathrm{id}}) \cap \mathrm{KUNodes}(\mathrm{BT}, \mathrm{RL})$. Let $S \cap W = \{\mathrm{att}_i\}_{i\in K}$. Since $|K| \geq t$, there exists a set $K' \subseteq K \cup J_2$ with size $d+1$. For all $j \in K'$, compute $r_{\theta,j} = \mathbf{e}_{\theta,j}^{\top}(\mathbf{c}; \mathbf{c}_\theta; \mathbf{c}_j)$, and $r_\theta = \sum_{j\in K'} L_j r_{\theta,j}$, where $L_j = (\prod_{k\in K', k\neq j}(-k)/\prod_{k\in K', k\neq j}(j-k))$. Finally, compute

$$M = \mathrm{Rec}(r_\theta, c_0) = \lfloor k \cdot \left(\frac{c_0}{g} - \frac{r_\theta}{q}\right)\rceil \bmod k. \tag{6}$$

4.2. Correctness. For $j \in K' \cap K$ and $\theta \in \mathrm{path}(\mathrm{BT}, \nu_{\mathrm{id}}) \cap \mathrm{KUNodes}$ (BT, RL), we have

$$\begin{aligned}\left(\mathbf{c}; \mathbf{c}_\theta; \mathbf{c}_j\right) &= \begin{bmatrix}\mathbf{c}\\ \mathbf{c}_\theta\\ \mathbf{c}_j\end{bmatrix} = \begin{bmatrix}\mathbf{A}^{\top}\mathbf{s} + D\mathbf{x}\\ \mathbf{D}_\theta^{\top}\cdot\mathbf{s} + D\mathbf{R}_\theta^{\top}\cdot\mathbf{x}\\ \left(\mathbf{B}_j + H\left(\mathrm{att}_j\right)\mathbf{G}\right)^{\top}\mathbf{s} + D\mathbf{R}_j^{\top}\mathbf{x}\end{bmatrix}\\ &= \left(\mathbf{A}\|\mathbf{D}_\theta\|\mathbf{B}_j + H\left(\mathrm{att}_j\right)\mathbf{G}\right)^{\top}\mathbf{s} + D\begin{bmatrix}\mathbf{x}\\ \mathbf{R}_\theta^{\top}\mathbf{x}\\ \mathbf{R}_j^{\top}\mathbf{x}\end{bmatrix}.\end{aligned} \tag{7}$$

For $j \in K' \cap J_2$ and $\theta \in \mathrm{path}(\mathrm{BT}, \nu_{\mathrm{id}}) \cap \mathrm{KUNodesBTRL}$, we have

$$\begin{aligned}\left(\mathbf{c}; \mathbf{c}_\theta; \mathbf{c}_j\right) &= \begin{bmatrix}\mathbf{c}\\ \mathbf{c}_\theta\\ \mathbf{c}_j\end{bmatrix} = \begin{bmatrix}\mathbf{A}^{\top}\mathbf{s} + D\mathbf{x}\\ \mathbf{D}_\theta^{\top}\cdot\mathbf{s} + D\mathbf{R}_\theta^{\top}\cdot\mathbf{x}\\ \left(\mathbf{B}_j + \mathbf{G}\right)^{\top}\mathbf{s} + D\mathbf{R}_j^{\top}\mathbf{x}\end{bmatrix}\\ &= \left(\mathbf{A}\|\mathbf{D}_\theta\|\mathbf{B}_j + \mathbf{G}\right)^{\top}\mathbf{s} + D\begin{bmatrix}\mathbf{x}\\ \mathbf{R}_\theta^{\top}\mathbf{x}\\ \mathbf{R}_j^{\top}\mathbf{x}\end{bmatrix}.\end{aligned} \tag{8}$$

Denote $\mathbf{x}_{\theta,j} = (\mathbf{x}; \mathbf{R}_\theta^{\top}\mathbf{x}; \mathbf{R}_j^{\top}\mathbf{x})$; then, $(\mathbf{c}; \mathbf{c}_\theta; \mathbf{c}_j) = \mathbf{E}_{\theta,j}^{\top}\mathbf{s} + D\mathbf{x}_{\theta,j}$ for both cases. Thus, we have $r_{\theta,j} = \mathbf{e}_{\theta,j}^{\top}\cdot(\mathbf{c}; \mathbf{c}_\theta; \mathbf{c}_j) = \mathbf{e}_{\theta,j}^{\top}\cdot(\mathbf{E}_{\theta,j}^{\top}\mathbf{s} + D\mathbf{x}_{\theta,j}) = \hat{\mathbf{u}}_{\theta,j}^{\top}\mathbf{s} + Dy_{\theta,j}$, where $y_{\theta,j} = \mathbf{e}_{\theta,j}^{\top}\cdot\mathbf{x}_{\theta,j}$. Hence, $r_\theta = \sum_{j\in K'} L_j r_{\theta,j} = \mathbf{u}^{\top}\mathbf{s} + y_\theta$, where $y_\theta = \sum_{j\in K'} D L_j y_{\theta,j}$. Finally, we have

$$\overline{d} = \left|\sigma_1 - \sigma_2\right| = \left|\left(\mathbf{u}^{\top}\mathbf{s} + Dx_0\right) - \left(\mathbf{u}^{\top}\mathbf{s} + y_\theta\right)\right| = \left|Dx_0 - y_\theta\right|. \tag{9}$$

Now, we begin to bound $|Dx_0 - y_\theta|$. By Lemmas 1 and 6, we have $\|\mathbf{e}_{\theta,j}\| \leq \sigma\sqrt{3m}$. Note that $\mathbf{e}_{\theta,j}^{\top}\cdot\mathbf{x}_{\theta,j} = \mathbf{e}_{\theta,j,0}^{\top}\cdot\mathbf{x} + \mathbf{e}_{\theta,j,1}^{\top}\cdot\mathbf{R}_\theta^{\top}\mathbf{x} + \mathbf{e}_{\theta,j,2}^{\top}\cdot\mathbf{R}_j^{\top}\mathbf{x}$, where $\mathbf{e}_{\theta,j}^{\top} = (\mathbf{e}_{\theta,j,0}^{\top}, \mathbf{e}_{\theta,j,1}^{\top}, \mathbf{e}_{\theta,j,2}^{\top})$. Since $\|\mathbf{e}_{\theta,j,0} + \mathbf{R}_\theta\mathbf{e}_{\theta,j,1} + \mathbf{R}_j\mathbf{e}_{\theta,j,2}\| \leq (s_1(\mathbf{R}_\theta) + s_1(\mathbf{R}_j) + 1)\cdot\sigma\sqrt{3m}$, by Lemma 4, we have $\mathbf{e}_{\theta,j}^{\top}\cdot\mathbf{x}_{\theta,j} \leq (s_1(\mathbf{R}_\theta) + s_1(\mathbf{R}_j) + 1)\cdot\sigma\sqrt{3m}\cdot(q\alpha\omega(\sqrt{\log m}) + \sqrt{m}/2)$. Applying Lemma 9 in [75], we have $DL_j \leq ((\ell+d)!)^4$. By Lemma 8, we have $s_1(\mathbf{R}_\theta) = O(\sqrt{m})$ and $s_1(\mathbf{R}_j) = O(\sqrt{m})$. Thus, $|y_\theta| \leq (d+1)((\ell+d)!)^4\sigma O(m)\cdot(q\alpha\omega(\sqrt{\log m}) + \sqrt{m}/2)$. Therefore, we have $|Dx_0 - y_\theta| \leq ((\ell+d)!)^2(q\alpha\omega(\sqrt{\log m}) + 1/2) + |y_\theta| \leq \sigma q\alpha m(d+1)((\ell+d)!)^4\omega(\sqrt{\log m}) + \sigma(d+1)(\ell+d!)^4 O(m^{3/2})$ by Lemma 4. According to Theorem 2, if $\overline{d} < (1/2)((q/k) - (q/g) - 1)$, then our scheme is correct.

4.3. Security. In this section, we prove the security of our construction of the DR-ABE scheme with user-level user revocation in the selective model in Definition 1. The proof is given in Appendix A.

Theorem 4. *For appropriate parameters n, m, q, σ, and α, the above DR-ABE scheme with user-level user revocation is secure provided that the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem is hard.*

4.4. Parameters. In this section, we will instantiate the parameters to satisfy the correctness and security of DR-ABE with user-level user revocation. In particular, we need to set parameters so that the following conditions hold with overwhelming possibility:

(i) For the algorithm TrapGen, we need $m \geq 2n\lceil\log q\rceil$ (i.e., Lemma 5)

(ii) For the algorithm SampleLeft, we need $\sigma \geq O(\sqrt{n\log q})\cdot\omega(\sqrt{\log m})$ (i.e., Lemma 5 and 6)

(iii) For correctness, we need $|Dx_0 - y_\theta| = \overline{d} < (1/2)((q/k) - (q/g) - 1)$

(iv) For security proof, we need $\sigma \geq \sqrt{m}\cdot\omega(\sqrt{\log m})$ for the algorithm SampleRight (i.e., Lemmas 5, 7, and 8) and $m > (n+1)\log q + \omega(\log n)$ (i.e., Lemma 9)

(v) For the hardness of LWE, we need $\alpha q > 2\sqrt{n}$ (i.e., Lemma 3)

Assume that δ is a real number such that $n^{1+\delta} > \lceil(n+1)\log q + \omega(\log n)\rceil$, and m, σ, q, and α are determined as follows:

(i) $m = 2n^{1+\delta}$.

(ii) $\sigma = \sqrt{m}\cdot\omega(\sqrt{\log m})$.

(iii) $q = \sigma m^{3/2}(d+1)((\ell+d)!)^4\omega(\sqrt{2\log m})\cdot(kg/(g-k))$.

(iv) $\alpha = (\sigma m(d+1)((\ell+d)!)^4\omega(\sqrt{\log m})\cdot(kg/(g-k)))^{-1}$.

5. DR-ABE with Attribute-Level Revocation

In this section, based on AKCN (Algorithm 1), we propose a DR-ABE scheme from lattices, achieving attribute-level user revocation and flexible threshold access policies on multi-valued attributes, which further illustrates the utility and versatility of KC/AKC.

5.1. Construction Details. The idea of constructing DR-ABE with user-level user revocation in Section 4 cannot be extended to constructing DR-ABE with attribute-level user revocation directly for the following reason. Suppose we associate every attribute att_i with a binary tree BT_i of depth L. For each id, we link id to a leaf node $v_{\text{id},i}$ of BT_i. Then, for each $l \in [L]$, the random vector $\mathbf{u}$ in the public key is secret-shared into vectors $\{\hat{\mathbf{u}}_{l,i}\}$, where $\hat{\mathbf{u}}_{l,i}$ is associated with the node of depth l in path$(\text{BT}, v_{\text{id},i})$ of BT_i. Now, if the non-revoked attribute set $S_{\text{id,RL}=\{\text{RL}_i\}} = \{\text{att}_i | \text{id} \notin \text{RL}_i\}$ of id satisfies the access structure, then $\mathbf{u}$ should be recovered if the extension works. Now, for each $\text{att}_i \in S_{\text{id,RL}}$, there exists $\theta_i \in \text{path}(\text{BT}, v_{\text{id},i}) \cap \text{KUNodes}(\text{BT}_i, \text{RL}_i)$, and thus, $\hat{\mathbf{u}}_{\theta_i,i}$ can be recovered. However, we cannot recover $\hat{\mathbf{u}}$ since θ_i may not be at the same depth.

The main ideas behind our construction can be described as follows. The random vector $\mathbf{u}$ in the public key is secret-shared into vectors $\{\hat{\mathbf{u}}_i\}$, where $\hat{\mathbf{u}}_i$ is associated with the i-th attribute att_i of the identity id. To revoke att_i of id, we further split each $\hat{\mathbf{u}}_i$ into two random vectors $\hat{\mathbf{u}}'_i$ and $\hat{\mathbf{u}}''_i$, corresponding to att_i and id, respectively. If att_i of id is revoked, $\hat{\mathbf{u}}''_i$, therefore, $\hat{\mathbf{u}}_i$ cannot be recovered. In this way, $\mathbf{u}$ can be recovered only if the set of nonrevoked attributes of id satisfies the threshold access policy, thereby achieving the revocation of part attributes of id.

For convenience, we use the notations from Section 4.

Setup$(n, \mathcal{R}, N)$: on inputting a security parameter n, a system attribute set $\mathcal{R} = \mathcal{R}_1 \times \cdots \times \mathcal{R}_\ell$, and a maximal number of users N in the system, this algorithm sets the primitive matrix $\mathbf{G}$ (with public trapdoor $\mathbf{T}_G$, see Lemma 5) and the parameters q, m, α, σ, k, g, and $\overline{d}$ as specified in Section 4.4. Then, it performs as follows:

(1) Run $(\mathbf{A}, \mathbf{T}_A) \longleftarrow \text{TrapGen}(n, m, q)$.
(2) Choose $\mathbf{B}_i \longleftarrow \mathbb{Z}_q^{n\times m}$ for $i \in \mathcal{I}$.
(3) Choose $\mathbf{u} \longleftarrow \mathbb{Z}_q^n$.
(4) Choose a full-rank difference map H: $\mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n\times n}$.
(5) Build a family of binary trees $\text{BT} = \{\text{BT}_i\}_{i\in\mathcal{I}_1}$, where each BT_i has N leaf nodes. For each $i \in \mathcal{I}_1$ and each node $\theta \in \text{BT}_i$, choose "identifier" $\mathbf{D}_{i,\theta} \longleftarrow \mathbb{Z}_q^{n\times m}$.
(6) Return $\text{PK} = \{\mathbf{A}, \{\mathbf{B}_i\}_{i\in,\mathcal{I}}, \mathbf{u}, H, \text{BT}\}$ and $\text{MSK} = \mathbf{T}_A$.

Keygen(PK, MSK, id, S): on inputting the public key PK, the master secret key MSK, an identity id, and the attribute set $S = \{\text{att}_i\}_{i\in I}$ of id, where $I \subseteq \mathcal{I}_1$ and $\text{att}_i \in \mathcal{R}_i$, it goes as follows:

(1) For $i \in [1, n]$, randomly choose degree d polynomial $p_i(x) \in \mathbb{Z}_q[x]$ such that $\mathbf{u} = (p_1(0), \ldots, p_n(0))^\top$. For each $i \in I \cup \mathcal{I}_2$, let $\hat{\mathbf{u}}_i = (p_1(i), \ldots, p_n(i))^\top$.
(2) For each $i \in I$, pick an unassigned leaf node $v_{\text{id},i}$ from BT_i and store id in that node. Choose $\hat{\mathbf{u}}'_i \longleftarrow \mathbb{Z}_q^n$ and set $\hat{\mathbf{u}}''_i = \hat{\mathbf{u}}_i - \hat{\mathbf{u}}'_i$. Sample vector $\mathbf{e}'_i \longleftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{B}_i + H(\text{att}_i)\mathbf{G}, \hat{\mathbf{u}}'_i, \mathbf{T}_A, \sigma)$. Sample $\mathbf{e}''_{i,\theta} \longleftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{D}_{i,\theta}, \hat{\mathbf{u}}''_i, \mathbf{T}_A, \sigma)$ for $\theta \in \text{path}(\text{BT}_i, v_{\text{id},i})$.
Let $\mathbf{E}'_i = (\mathbf{A} \| \mathbf{B}_i + H(\text{att}_i)\mathbf{G})$ and $\mathbf{E}''_{i,\theta} = (\mathbf{A} \| \mathbf{D}_{i,\theta})$; note that $\mathbf{E}'_i \cdot \mathbf{e}'_i = \hat{\mathbf{u}}'_i$ and $\mathbf{E}''_i \cdot \mathbf{e}''_{i,\theta} = \hat{\mathbf{u}}''_i$.
(3) For each $i \in \mathcal{I}_2$, sample $(\mathbf{e}_i \longleftarrow \text{SampleLeft}\mathbf{A}, \mathbf{B}_i + \mathbf{G}, \hat{\mathbf{u}}_i, \mathbf{T}_A, \sigma)$. Let $\mathbf{E}_i = (\mathbf{A} \| \mathbf{B}_i + \mathbf{G})$; note that $\mathbf{E}_i \cdot \mathbf{e}_i = \hat{\mathbf{u}}_i$.
(4) Return $\text{sk}_{S,\text{id}} = (\{\mathbf{e}'_i\}_{i\in I}, \{\mathbf{e}''_{i,\theta}\}_{i\in I, \theta\in\text{path}(\text{BT}_i, v_{\text{id},i})}, \{\mathbf{e}_i\}_{i\in,\mathcal{I}_2})$ as the private key.

Note that, for any subset $K \subseteq I \cup \mathcal{I}_2$, $|K| = d + 1$, we have $\mathbf{u} = \sum_{i\in K} L_i \cdot \hat{\mathbf{u}}_i$, where the Lagrange coefficient $L_i = (\prod_{j\in K, j\neq i}(-j)/\prod_{j\in K, j\neq i}(i - j))$.

Enc(PK, (W, t), RL, M): on inputting a public key PK, an attribute set $W = \{\text{att}_j\}_{j\in J_1}$, an integer $1 \le t \le \min(|W|, d)$, a family of attribute revocation lists $\text{RL} = \{\text{RL}_j\}_{j\in J_1}$, where each RL_j consists of identities whose j-th attribute is revoked, and a message $M \in \mathbb{Z}_k$, it works as follows:

(1) Choose $\mathbf{s} \longleftarrow \mathbb{Z}_q^n$ and compute

$$c_0 = \text{Con}(\mathbf{u}^\top \mathbf{s} + Dx_0, M) = \lfloor \frac{g \cdot (\mathbf{u}^\top \mathbf{s} + Dx_0 + \lfloor Mq/k \rfloor)}{q} \rceil \bmod g, \tag{10}$$

and $\mathbf{c} = \mathbf{A}^\top \mathbf{s} + D\mathbf{x}$, where $x_0 \longleftarrow \overline{\Psi}_\alpha$ and $\mathbf{x} \longleftarrow \overline{\Psi}_\alpha^m$.
(2) For each $j \in J_1$, choose $\mathbf{R}_j \longleftarrow \{-1, 1\}^{m\times m}$ and compute $\mathbf{c}'_j = (\mathbf{B}_j + H(\text{att}_j)\mathbf{G})^\top \mathbf{s} + D\mathbf{R}_j^\top \mathbf{x}$.
(3) For each $j \in J_1$ and each $\theta \in \text{KUNodesBT}_j\text{RL}_j$, choose $\mathbf{R}_{j,\theta} \longleftarrow \{-1, 1\}^{m\times m}$ and compute $\mathbf{c}''_{j,\theta} = \mathbf{D}_{j,\theta}^\top \cdot \mathbf{s} + D\mathbf{R}_{j,\theta}^\top \cdot \mathbf{x}$.
(4) Let $J_2 = \{\ell + 1, \ldots, \ell + d + 1 - t\}$, and for each $j \in J_2$, choose $\mathbf{R}_j \longleftarrow \{-1, 1\}^{m\times m}$ and compute $\mathbf{c}_j = (\mathbf{B}_j + \mathbf{G})^\top \mathbf{s} + D\mathbf{R}_j^\top \mathbf{x}$.
(5) Return $C = (c_0, \mathbf{c}, \{\mathbf{c}'_j\}_{j\in J_1}, \{\mathbf{c}''_{j,\theta}\}_{j\in J_1, \theta\in\text{KUNodesBT}_j\text{RL}_j}, \{\mathbf{c}_j\}_{j\in J_2})$ as the ciphertext.

Dec(PK, $\text{sk}_{S,\text{id}}$, C): on input the public key PK, the private key $\text{sk}_{S,\text{id}}$ a ciphertext C, it works as follows. Let $S_{\text{id,RL}} = \{\text{att}_i \in S | \text{id} \notin \text{RL}_i, i \in I\}$.

(1) If $|S_{\text{id,RL}} \cap W| < t$, then return $\perp$.
(2) Else, parse $\text{sk}_{S,\text{id}} = (\{\mathbf{e}'_i\}_{i\in I}, \{\mathbf{e}''_{i,\theta}\}_{i\in I, \theta\in\text{path}(\text{BT}_i, v_{\text{id},i})}, \{\mathbf{e}_i\}_{i\in,\mathcal{I}_2})$ and $C = (c_0, \mathbf{c}, \{\mathbf{c}'_j\}_{j\in J_1}, \{\mathbf{c}''_{j,\theta}\}_{j\in J_1, \theta\in\text{KUNodesBT}_j\text{RL}_j}, \{\mathbf{c}_j\}_{j\in J_2})$. Let $S_{\text{id,RL}} \cap W = \{\text{att}_i\}_{i\in K}$. Since $|K| \ge t$, there exists a set $K' \subseteq K \cup J_2$ with size $d+1$. For all $j \in K' \cap K$, there exists $\theta_j \in \text{path}(\text{BT}_j, v_{\text{id},j}) \cap \text{KUNodes}(\text{BT}_i, \text{RL}_j)$, and compute $r_j = \mathbf{e}'^\top_j(\mathbf{c}; \mathbf{c}'_j) + \mathbf{e}''^\top_{j,\theta_j}(\mathbf{c}; \mathbf{c}''_{j,\theta_j})$. For all $j \in K' \cap J_2$, compute $r_j = \mathbf{e}_j^\top(\mathbf{c}; \mathbf{c}_j)$. Then, compute $r = \sum_{j\in K'} L_j r_j$,

where $L_j = (\prod_{k\in K', k\neq j}(-k)/\prod_{k\in K', k\neq j}(j-k))$. Finally, compute

$$M = \mathrm{Rec}(r, c_0) = \lfloor k \cdot \left(\frac{c_0}{g} - \frac{r}{q}\right) \rceil \bmod k. \tag{11}$$

5.2. Correctness. For $j \in K' \cap K$ and $\theta_j \in \mathrm{path}(\mathrm{BT}_j, \gamma_{\mathrm{id},j}) \cap \mathrm{KUNodes}(\mathrm{BT}_j, \mathrm{RL}_j)$, we have

$$\begin{aligned}
\left(\mathbf{c}; \mathbf{c}'_j\right) &= \begin{bmatrix} \mathbf{c} \\ \mathbf{c}'_j \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top \mathbf{s} + D\mathbf{x} \\ \left(\mathbf{B}_j + H\left(\mathrm{att}_j\right)\mathbf{G}\right)^\top \mathbf{s} + D\mathbf{R}_j^\top \mathbf{x} \end{bmatrix} \\
&= \left(\mathbf{A} \| \mathbf{B}_j + H\left(\mathrm{att}_j\right)\mathbf{G}\right)^\top \mathbf{s} + D\begin{bmatrix} \mathbf{x} \\ \mathbf{R}_j^\top \mathbf{x} \end{bmatrix}, \\
\left(\mathbf{c}; \mathbf{c}''_{j,\theta_j}\right) &= \begin{bmatrix} \mathbf{c} \\ \mathbf{c}''_{j,\theta_j} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top \mathbf{s} + D\mathbf{x} \\ \mathbf{D}_{j,\theta_j}^\top \cdot \mathbf{s} + D\mathbf{R}_{j,\theta_j}^\top \cdot \mathbf{x} \end{bmatrix} \\
&= \left(\mathbf{A} \| \mathbf{D}_{j,\theta_j}\right)^\top \mathbf{s} + D\begin{bmatrix} \mathbf{x} \\ \mathbf{R}_{j,\theta_j}^\top \mathbf{x} \end{bmatrix}.
\end{aligned} \tag{12}$$

Thus,

$$\begin{aligned}
r_j &= \mathbf{e}_j'^\top \left(\mathbf{c}; \mathbf{c}'_j\right) + \mathbf{e}''^\top_{j,\theta_j} \left(\mathbf{c}; \mathbf{c}''_{j,\theta_j}\right) \\
&= \mathbf{e}_j'^\top \left(\mathbf{A} \| \mathbf{B}_i + H\left(\mathrm{att}_i\right)\mathbf{G}\right)^\top \mathbf{s} + D\mathbf{e}_j'^\top \left(\mathbf{x}; \mathbf{R}_j^\top \mathbf{x}\right) \\
&\quad + \mathbf{e}''^\top_{j,\theta_j} \left(\mathbf{A} \| \mathbf{D}_{j,\theta_j}\right)^\top \mathbf{s} + D\mathbf{e}''^\top_{j,\theta_j} \left(\mathbf{x}; \mathbf{R}_{j,\theta_j}^\top\right) \\
&= \left(\widehat{\mathbf{u}}_j'^\top + \widehat{\mathbf{u}}_j''^\top\right)\mathbf{s} + D\left(\mathbf{e}_j'^\top \mathbf{x}'_j + \mathbf{e}''^\top_{j,\theta_j} \mathbf{x}''_{j,\theta_j}\right) \\
&= \widehat{\mathbf{u}}_j^\top \mathbf{s} + D\left(\mathbf{e}_j'^\top \mathbf{x}'_j + \mathbf{e}''^\top_{j,\theta_j} \mathbf{x}''_{j,\theta_j}\right),
\end{aligned} \tag{13}$$

where $\mathbf{x}'_j = (\mathbf{x}; \mathbf{R}_j^\top \mathbf{x})$ and $\mathbf{x}''_{j,\theta_j} = (\mathbf{x}; \mathbf{R}_{j,\theta_j}^\top \mathbf{x})$.

For $j \in K' \cap J_2$, we have

$$\begin{aligned}
\left(\mathbf{c}; \mathbf{c}_j\right) &= \begin{bmatrix} \mathbf{c} \\ \mathbf{c}_j \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top \mathbf{s} + D\mathbf{x} \\ \left(\mathbf{B}_j + \mathbf{G}\right)^\top \mathbf{s} + D\mathbf{R}_j^\top \mathbf{x} \end{bmatrix} \\
&= \left(\mathbf{A} \| \mathbf{B}_j + \mathbf{G}\right)^\top \mathbf{s} + D\begin{bmatrix} \mathbf{x} \\ \mathbf{R}_j^\top \mathbf{x} \end{bmatrix}.
\end{aligned} \tag{14}$$

Thus, $r_j = \mathbf{e}_j^\top (\mathbf{c}; \mathbf{c}_j) = \mathbf{e}_j^\top (\mathbf{A} \| \mathbf{B}_j + \mathbf{G})^\top \mathbf{s} + D\mathbf{e}_j^\top (\mathbf{x}; \mathbf{R}_j^\top \mathbf{x}) = \widehat{\mathbf{u}}_j^\top \mathbf{s} + D\mathbf{e}_j^\top \mathbf{x}'_j$, where $\mathbf{x}'_j = (\mathbf{x}; \mathbf{R}_j^\top \mathbf{x})$.

Then, we have

$$\begin{aligned}
r &= \sum_{j\in K'} L_j r_j \\
&= \sum_{j\in K' \cap K} L_j \left(\widehat{\mathbf{u}}_j^\top \mathbf{s} + D\left(\mathbf{e}_j'^\top \mathbf{x}'_j + \mathbf{e}''^\top_{j,\theta_j} \mathbf{x}''_{j,\theta_j}\right)\right) \\
&\quad + \sum_{j\in K' \cap J_2} L_j \left(\widehat{\mathbf{u}}_j^\top \mathbf{s} + D\mathbf{e}_j^\top \mathbf{x}'_j\right) \\
&= \left(\sum_{j\in K'} L_j \widehat{\mathbf{u}}_j^\top\right)\mathbf{s} + y = \mathbf{u}^\top \mathbf{s} + y,
\end{aligned} \tag{15}$$

where $y = D(\sum_{j\in K'} L_j \mathbf{e}_j'^\top \mathbf{x}'_j + \sum_{j\in K'\cap K} L_j \mathbf{e}''^\top_{j,\theta_j} \mathbf{x}''_{j,\theta_j})$.

Finally, we have

$$\overline{d} = |\sigma_1 - \sigma_2| = |(\mathbf{u}^\top \mathbf{s} + Dx_0) - (\mathbf{u}^\top \mathbf{s} + y)| = |Dx_0 - y|. \tag{16}$$

Now, we begin to bound $|Dx_0 - y|$. By Lemmas 1 and 6, we have $\|\mathbf{e}'_j\| \le \sigma\sqrt{2m}$ and $\|\mathbf{e}''_{j,\theta_j}\| \le \sigma\sqrt{2m}$. For $j \in K'$, $\mathbf{e}_j'^\top \mathbf{x}'_j = \mathbf{e}_{j,0}'^\top \mathbf{x} + \mathbf{e}_{j,1}'^\top \mathbf{R}_j^\top \mathbf{x}$, where $\mathbf{e}'_j = (\mathbf{e}_{j,0}'; \mathbf{e}_{j,1}')$. Since $\|\mathbf{e}_{j,0}' + \mathbf{R}_j \mathbf{e}_{j,1}'\| \le (s_1(\mathbf{R}_j) + 1) \cdot \sigma\sqrt{2m}$, by Lemma 4, we have $|\mathbf{e}_j'^\top \mathbf{x}'_j| \le (s_1(\mathbf{R}_j) + 1)\sigma\sqrt{2m}(q\alpha\omega(\sqrt{\log m}) + \sqrt{m}/2)$. Similarly, for $j \in K' \cap K$, we have $|\mathbf{e}''^\top_{j,\theta_j} \mathbf{x}''_{j,\theta_j}| \le (s_1(\mathbf{R}_{j,\theta_j}) + 1)\sigma\sqrt{2m}(q\alpha\omega(\sqrt{\log m}) + \sqrt{m}/2)$. Applying Lemma 9 in [75], we have $DL_j \le ((\ell + d)!)^4$. By Lemma 8, we have $s_1(\mathbf{R}_j), s_1(\mathbf{R}_{j,\theta_j}) = O(\sqrt{m})$. Thus, $|y| \le 2(d+1)((\ell+d)!)^4 \sigma O(m) \cdot (q\alpha\omega(\sqrt{\log m}) + \sqrt{m}/2)$. Therefore, we have $|Dx_0 - y| \le ((\ell+d)!)^2 (q\alpha\omega(\sqrt{\log m}) + 1/2) + |y| \le \sigma q\alpha m(d+1)((\ell+d)!)^4 \omega(\sqrt{\log m}) + \sigma(d+1)((\ell+d)!)^4 O(m^{3/2})$ by Lemma 4. According to Theorem 2, if $\overline{d} < (1/2)((q/k) - (q/g) - 1)$, then our scheme is correct.

5.3. Security. In this section, we prove the security of our DR-ABE scheme with attribute-level user revocation. The proof is given in Appendix B.

Theorem 5. *For appropriate parameters n, m, q, σ, and α, the above DR-ABE scheme with attribute-level user revocation is secure provided that the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem is hard.*

5.4. Parameters. The parameters are the same as those of Section 4.4.

6. Conclusion

In this work, we demonstrate the power of KC/AKC by proposing two special types of PKE schemes. Specifically, on the basis of AKC, combined with PKE/KEM protocols submitted to the NIST, FRD, trapdoor for lattices, Gaussian sampling, leftover hash lemma, and the binary tree structure, we propose two special kinds of PKE schemes, i.e., directly revocable ciphertext-policy attribute-based encryption schemes from LWE. One achieves user-level user revocation, while the other achieves attribute-level user revocation. Both schemes inherit the main advantages of the direct revocation mechanism: the revocation list is defined by the message sender; the authority does not need to generate and issue key update anymore. In addition, both schemes support multibit encryption and flexible threshold access policies on multivalued attributes. The size of the public key of our schemes can be reduced in the random oracle model. Most parts of the decryption work can be outsourced to a third party as well. Our schemes proved to be secure against quantum attacks in the standard model, assuming the hardness of the LWE problem. The two schemes imply the versatility of KC/AKC. Compared with other existing lattice-based revocable CP-ABE schemes, our schemes have reasonable security guarantee.

Appendix

A. Proof of Theorem 4

Proof. Suppose there exists a PPT adversary $\mathscr{A}$ which breaks the security of our DR-ABE scheme with user-level user revocation with nonnegligible probability, we can construct an algorithm $\mathscr{B}$ that solves the LWE problem with the same advantage.

Note that $\mathscr{B}$ has an oracle $\mathcal{O}(\cdot)$, and he wants to determine whether it is a noisy pseudo-random sampler $\mathcal{O}_{\mathbf{s}^*}$ for some $\mathbf{s}^* \in \mathbb{Z}_q^n$ or a truly random sampler $\mathcal{O}_{\$}$. To this end, $\mathscr{B}$ proceeds as follows:

Init: $\mathscr{A}$ submits a challenge access structure ÅA* = $(\mathrm{W}^* = \{\mathrm{att}_j^*\}_{\mathrm{j}\in \mathrm{J}_1^*}, \mathrm{t}^*)$ and a challenge revocation list RL* to $\mathscr{B}$, where $J_1^* \subseteq \mathscr{I}_1$ and $1 \le t^* \le \min(|W^*|, d)$. Let $J_2^* = \{\ell + 1, \ldots, \ell + d + 1 - t^*\}$ and $J^* = J_1^* \cup J_2^*$.

Setup: after receiving $(W^* = \{\mathrm{att}_j^*\}_{j\in J^*}, t^*)$ and RL*, $\mathscr{B}$ samples $(\mathbf{u}, v_u) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ and $(\mathbf{A}, \mathbf{v}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$ from $\mathcal{O}(\cdot)$, chooses an FRD map $H: \mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n\times n}$, and builds a binary tree BT with N leaf nodes.

(i) For each $j \in J_1^*$, $\mathscr{B}$ chooses $\mathbf{R}_j^* \longleftarrow \{-1, 1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{AR}_j^* - H(\mathrm{att}_j^*)\mathbf{G}$

(ii) For each $j \in \mathscr{I}_1 \backslash J_1^*$, $\mathscr{B}$ chooses $\mathbf{R}_j^* \longleftarrow \{-1, 1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{AR}_j^* - H(0)\mathbf{G}$

(iii) For each $j \in J_2^*$, $\mathscr{B}$ chooses $\mathbf{R}_j^* \longleftarrow \{-1, 1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{AR}_j^* - \mathbf{G}$

(iv) For each $j \in \mathscr{I}_2 \backslash J_2^*$, $\mathscr{B}$ chooses $\mathbf{R}_j^* \longleftarrow \{-1, 1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{AR}_j^*$

(v) For each $\theta \in$ BT, $\mathscr{B}$ chooses $\mathbf{R}_\theta^* \longleftarrow \{-1, 1\}^{m\times m}$ and computes $\mathbf{D}_\theta = \mathbf{AR}_\theta^*$ if $\theta \in$ KUNodes (BT, RL*), and $\mathbf{D}_\theta = \mathbf{AR}_\theta^* + \mathbf{G}$, otherwise

Finally, $\mathscr{B}$ sends the public key PK = $\{\mathbf{A}, \{\mathbf{B}_i\}_{i\in\mathscr{I}}, \mathbf{u}, H, \mathrm{BT}\}$ to $\mathscr{A}$ and keeps $(\{\mathbf{R}_j^*\}_{j\in\mathscr{I}}, \{\mathbf{R}_\theta^*\}_{\theta\in\mathrm{BT}}, v_u, \mathbf{v})$ secret.

Phases 1 and 2: when $\mathscr{B}$ receives a key generation query (id, S) from $\mathscr{A}$, where $S = \{\mathrm{att}_i\}_{i\in I}$, he outputs $\perp$ if $S \vDash (W^*, t^*)$ and id $\notin$ RL*. Otherwise, $\mathscr{B}$ picks an unassigned leaf node v_{id} from BT and stores id in that node.

(i) For id $\in$ RL*, note that, in this case, $\mathrm{path}(\mathrm{BT}, v_{\mathrm{id}}) \cap \mathrm{KUNodes}(\mathrm{BT}, \mathrm{RL}^*) = \varnothing$. For each node $\theta \in \mathrm{path}(\mathrm{BT}, v_{\mathrm{id}})$, $\mathscr{B}$ first picks n degree d polynomials $p_{\theta,1}(x), \ldots, p_{\theta,n}(x) \in \mathbb{Z}_q[x]$ such that $\mathbf{u} = (p_{\theta,1}(0), \ldots, p_{\theta,n}(0))^\top$. Then, for each $i \in I \cup \mathscr{I}_2$, $\mathscr{B}$ sets $\hat{\mathbf{u}}_{\theta,i} = (p_{\theta,1}(i), \ldots, p_{\theta,n}(i))^\top$. Note that $\mathbf{E}_{\theta,i} = (\mathbf{A}\|\mathbf{AR}_\theta^* + \mathbf{G}\|\mathbf{B}_i + H(\mathrm{att}_i)\mathbf{G})$ for $i \in I$ and $\mathbf{E}_{\theta,i} = (\mathbf{A}\|\mathbf{AR}_\theta^* + \mathbf{G}\|\mathbf{B}_i + \mathbf{G})$ for $i \in \mathscr{I}_2$. Now, for each $\theta \in \mathrm{path}(\mathrm{BT}, v_{\mathrm{id}})$ and each $i \in I \cup \mathscr{I}_2$, $\mathscr{B}$ first chooses $\mathbf{e}_{\theta,i}'' \longleftarrow \mathscr{D}_{\mathbb{Z}^m}\sigma$, computes $\hat{\mathbf{u}}_{\theta,i}'' = (\mathbf{B}_i + H(\mathrm{att}_i)\mathbf{G}) \cdot \mathbf{e}_{\theta,i}''$ if $i \in I$ and $\hat{\mathbf{u}}_{\theta,i}'' = (\mathbf{B}_i + \mathbf{G}) \cdot \mathbf{e}_{\theta,i}''$ if $i \in \mathscr{I}_2$, runs $\mathbf{e}_{\theta,i}' \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}_\theta^*, \hat{\mathbf{u}}_{\theta,i}', \mathbf{T}_G, \sigma)$, where $\hat{\mathbf{u}}_{\theta,i}' = \hat{\mathbf{u}}_{\theta,i} - \hat{\mathbf{u}}_{\theta,i}''$, and then sets $\mathbf{e}_{\theta,i} = (\mathbf{e}_{\theta,i}' \| \mathbf{e}_{\theta,i}'')$.

(ii) For id $\notin$ RL* and $S \not\models (W^*, t^*)$, there exists $\theta^* \in \mathrm{path}(\mathrm{BT}, v_{\mathrm{id}}) \cap \mathrm{KUNodes}(\mathrm{BT}, \mathrm{RL}^*)$. For each $\theta \in \mathrm{path}(\mathrm{BT}, v_{\mathrm{id}}) \backslash \{\theta^*\}$, $\mathscr{B}$ picks n degree d polynomials $p_{\theta,1}(x), \ldots, p_{\theta,n}(x)$ such that $\mathbf{u} = (p_{\theta,1}(0), \ldots, p_{\theta,n}(0))^\top$. Then, it sets $\hat{\mathbf{u}}_{\theta,i} = (p_{\theta,1}(i), \ldots, p_{\theta,n}(i))^\top$ and generates $\mathbf{e}_{\theta,i} = (\mathbf{e}_{\theta,i}' \| \mathbf{e}_{\theta,i}'')$ for $i \in I \cup \mathscr{I}_2$ by using the Gaussian sampling and the SampleRight algorithms according to the above process.

For θ^*, let $S \cap W^* = \{\mathrm{att}_j\}_{j\in K}$; then, $|K| < t^*$. Thus, $|K \cup J_2^*| \le d$. $\mathscr{B}$ chooses a set K' such that $K \cup J_2^* \subseteq K' \subseteq I \cup \mathscr{I}_2$ and $|K'| = d$. For each $i \in K'$, $\mathscr{B}$ chooses $\mathbf{e}_{\theta^*,i} \longleftarrow \mathscr{D}_{\mathbb{Z}^{3m}}\sigma$, and if $i \in I$, let $\mathbf{E}_{\theta^*,i} = (\mathbf{A}\|\mathbf{D}_{\theta^*}\|\mathbf{B}_i + H(\mathrm{att}_i)\mathbf{G})$; else, let $\mathbf{E}_{\theta^*,i} = (\mathbf{A}\|\mathbf{D}_{\theta^*}\|\mathbf{B}_i + \mathbf{G})$. Then, $\mathscr{B}$ computes $\hat{\mathbf{u}}_{\theta^*,i} = \mathbf{E}_{\theta^*,i} \cdot \mathbf{e}_{\theta^*,i}$. Thus, we have $d + 1$ n-dimensional vectors $\{\mathbf{u}, \{\hat{\mathbf{u}}_{\theta^*,i}\}_{i\in K'}\}$. By the Lagrange interpolation formula, we can recover polynomials $p_{\theta^*,1}(x), \ldots, p_{\theta^*,n}(x)$ such that $\mathbf{u} = (p_{\theta^*,1}(0), \ldots, p_{\theta^*,n}(0))^\top$, and for each $i \in K'$, $\hat{\mathbf{u}}_{\theta^*,i} = (p_{\theta^*,1}(i), \ldots, p_{\theta^*,n}(i))^\top$. Now, for each $i \in I \backslash (K' \cap I)$, if $i \in J_1^*$, we have $\mathrm{att}_i \neq \mathrm{att}_i^*$ and $\mathbf{E}_{\theta^*,i} = (\mathbf{A}\|\mathbf{D}_{\theta^*}\|\mathbf{AR}_i^* + (H(\mathrm{att}_i) - H(\mathrm{att}_i^*))\mathbf{G})$; else, we have $\mathrm{att}_i \neq \mathbf{0}$ and $\mathbf{E}_{\theta^*,i} = (\mathbf{A}\|\mathbf{D}_{\theta^*}\|\mathbf{AR}_i^* + (H(\mathrm{att}_i) - H(0))\mathbf{G})$. For each $i \in \mathscr{I}_2 \backslash (K' \cap \mathscr{I}_2)$, note that we have $\mathbf{E}_{\theta^*,i} = (\mathbf{A}\|\mathbf{D}_{\theta^*}\|\mathbf{AR}_i^* + \mathbf{G})$. Now, for $i \in I \cup \mathscr{I}_2$, $\mathscr{B}$ first chooses $\mathbf{e}_{\theta^*,i}'' \longleftarrow \mathscr{D}_{\mathbb{Z}^m}\sigma$, computes $\hat{\mathbf{u}}_{\theta^*,i}'' = \mathbf{D}_{\theta^*} \cdot \mathbf{e}_{\theta^*,i}''$ and $\hat{\mathbf{u}}_{\theta^*,i}' = \hat{\mathbf{u}}_{\theta^*,i} - \hat{\mathbf{u}}_{\theta^*,i}''$, then runs $(\mathbf{e}_{\theta^*,i}', \mathbf{e}_{\theta^*,i}'') \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}_i^*\, \hat{\mathbf{u}}_{\theta^*,i}', \mathbf{T}_G, \sigma)$, and sets $\mathbf{e}_{\theta^*,i} = (\mathbf{e}_{\theta^*,i}' \| \mathbf{e}_{\theta^*,i}'' \| \mathbf{e}_{\theta^*,i}'')$.

In the end, $\mathcal{B}$ returns $\mathrm{sk}_{S,\mathrm{id}} = (\{\mathbf{e}_{\theta,i}\}_{\theta\in\mathrm{path}(\mathrm{BT},\nu_{\mathrm{id}}),i\in I\cup\mathcal{J}_2})$ to $\mathcal{A}$.

Challenge: when $\mathcal{A}$ submits two different messages $M_0, M_1 \in \mathbb{Z}_k$, the adversary $\mathcal{B}$ picks $b \in \{0,1\}$ and computes $c_0 = \mathrm{Con}(D\nu_u, M_b)$, $\mathbf{c} = D\mathbf{v}$. Then, $\mathcal{B}$ computes $\mathbf{c}_j = D(\mathbf{R}_j^*)^\top\mathbf{v}$ for each $j \in J_1^* \cup J_2^*$ and $\mathbf{c}_\theta = D(\mathbf{R}_\theta^*)^\top\mathbf{v}$ for each $\theta \in \mathrm{KUNodes}(\mathrm{BT}, \mathrm{RL}^*)$. Finally, $\mathcal{B}$ sends to $\mathcal{A}$ the ciphertext $C = (c_0, \mathbf{c}, \{\mathbf{c}_j\}_{j\in J_1^*\cup J_2^*}, \{\mathbf{c}_\theta\}_{\theta\in\mathrm{KUNodes}(\mathrm{BT},\mathrm{RL}^*)})$.

Guess: $\mathcal{A}$ output a guess $b' \in \{0,1\}$ for b. If $b' = b$, $\mathcal{B}$ outputs 1; else, $\mathcal{B}$ outputs 0.

Note that, by Lemma 3, the pair $(\mathbf{A}, \mathbf{u})$ is computationally indistinguishable from its distribution in the real attack. Applying Lemma 9, we know that $\{\mathbf{B}_i\}_{i\in\mathcal{J}}$ and $\{\mathbf{D}_\theta\}_{\theta\in\mathrm{BT}}$ are statistically close to uniform even given more information about $(\mathbf{R}_i^*)^\top\mathbf{x}$ and $(\mathbf{R}_\theta^*)^\top\mathbf{x}$, respectively. Hence, the distribution of the public key in the simulation is indistinguishable from that in the real attack, and $\mathcal{A}$ gains negligible information about $\{\mathbf{R}_i^*\}_{i\in\mathcal{J}}$ and $\{\mathbf{R}_\theta^*\}_{\theta\in\mathrm{BT}}$ from the public key. According to Lemmas 2, 6, and 7, the output distribution of the key generation simulation using the SampleRight algorithm is statistical to that in the real attack.

If $\mathcal{O}(\cdot) = \mathcal{O}_{s^*}$ for some $\mathbf{s}^*$, we claim that the challenge ciphertext C^* is a valid ciphertext for $\mathbf{s} = D\mathbf{s}^*$, $\{\mathbf{R}_i^*\}_{i\in J_1^*\cup J_2^*}$, and $\{\mathbf{R}_\theta^*\}_{\theta\in\mathrm{KUNodes}(\mathrm{BT}_j,\mathrm{RL}_j^*)}$: note that, for each $j \in J_1^*$, $\mathbf{c}_j = D(\mathbf{R}_j^*)^\top\cdot(\mathbf{A}^\top\mathbf{s}^* + \mathbf{x}) = (\mathbf{A}\mathbf{R}_j^*)^\top\cdot(D\mathbf{s}^*) + D\cdot(\mathbf{R}_j^*)^\top\mathbf{x} = (B_j + H(\mathrm{att}_j^*)\mathbf{G})^\top\mathbf{s} + D(\mathbf{R}_j^*)^\top\mathbf{x}$. For each $j \in J_2^*$, $\mathbf{c}_j = D(\mathbf{R}_j^*)^\top\cdot(\mathbf{A}^\top\mathbf{s}^* + \mathbf{x}) = (\mathbf{A}\mathbf{R}_j^*)^\top\cdot(D\mathbf{s}^*) + D\cdot(\mathbf{R}_j^*)^\top\mathbf{x} = (\mathbf{B}_j + \mathbf{G})^\top\mathbf{s} + D(\mathbf{R}_j^*)^\top\mathbf{x}$. For each $\theta \in \mathrm{KUNodesBTRL}^*$, $\mathbf{c}_\theta = D(\mathbf{R}_\theta^*)^\top(\mathbf{A}^\top\mathbf{s}^* + \mathbf{x}) = (\mathbf{A}\mathbf{R}_\theta^*)^\top(D\mathbf{s}^*) + D(\mathbf{R}_\theta^*)^\top\mathbf{x} = \mathbf{D}_\theta^\top\cdot\mathbf{s} + D(\mathbf{R}_\theta^*)^\top\mathbf{x}$. Therefore, the ciphertext is the same as the view of $\mathcal{A}$ in the real attack. Hence, if $\mathcal{A}$ guesses right b with noticeable probability more than 1/2, then $\mathcal{B}$ can succeed in its game with the same probability. Else, if $\mathcal{O}(\cdot) = \mathcal{O}_\$$, by Theorem 3, c_0 and M_b are independent. Since M_b is uniformly distributed, the probability of $\mathcal{A}$ guesses right b is exactly 1/2. In a word, if $\mathcal{A}$ breaks the security of our DR-ABE with user-level user revocation, then $\mathcal{B}$ solves the underlying LWE problem. □

B. Proof of Theorem 5

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ which breaks the security of our DR-ABE scheme with nonnegligible probability, we can construct an algorithm $\mathcal{B}$ that solves the LWE problem with the same advantage.

Note that $\mathcal{B}$ has an oracle $\mathcal{O}(\cdot)$, and he wants to determine whether it is a noisy pseudo-random sampler $\mathcal{O}_{\mathbf{s}^*}$ for some $\mathbf{s}^* \in \mathbb{Z}_q^n$ or a truly random sampler $\mathcal{O}_\$$. To this end, $\mathcal{B}$ proceeds as follows:

Init: $\mathcal{A}$ submits a challenge access structure $\text{ÅA}^* = (\mathrm{W}^* = \{\mathrm{att}_j^*\}_{\mathrm{j}\in\mathrm{J}^*}, \mathrm{t}^*)$ and a family of challenge attribute revocation lists $\mathrm{RL}^* = \{\mathrm{RL}_j^*\}_{j\in J_1^*}$ to $\mathcal{B}$, where $J_1^* \subseteq \mathcal{J}_1$ and $1 \le t^* \le \min(|W^*|, d)$. Let $J_2^* = \{\ell+1, \ldots, \ell+d+1-t^*\}$ and $J^* = J_1^* \cup J_2^*$.

Setup: after receiving $(W^* = \{\mathrm{att}_j^*\}_{j\in J_1^*}, t^*)$ and RL^*, $\mathcal{B}$ samples $(\mathbf{u}, \nu_u) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ and $(\mathbf{A}, \mathbf{v}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$ from $\mathcal{O}(\cdot)$, chooses an FRD map H: $\mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n\times n}$, and builds a family of binary trees $\mathrm{BT} = \{\mathrm{BT}_i\}_{i\in\mathcal{J}_1}$, where each BT_i has N leaf nodes.

(i) For each $j \in J_1^*$ and each $\theta \in \mathrm{BT}_j$, $\mathcal{B}$ randomly chooses $\mathbf{R}_j^*, \mathbf{R}_{j,\theta}^* \longleftarrow \{-1,1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{A}\mathbf{R}_j^* - H(\mathrm{att}_j^*)\mathbf{G}$ and $\mathbf{D}_{j,\theta} = \mathbf{A}\mathbf{R}_{j,\theta}^*$ if $\theta \in \mathrm{KUNodes}(\mathrm{BT}_j, \mathrm{RL}_j^*)$ and $\mathbf{D}_{j,\theta} = \mathbf{A}\mathbf{R}_{j,\theta}^* + \mathbf{G}$ if $\theta \in \mathrm{BT}_j \setminus \mathrm{KUNodes}(\mathrm{BT}_j, \mathrm{RL}_j^*)$

(ii) For each $j \in \mathcal{J}_1 \setminus J_1^*$ and each $\theta \in \mathrm{BT}_j$, $\mathcal{B}$ randomly chooses $\mathbf{R}_j^*, \mathbf{R}_{j,\theta}^* \longleftarrow \{-1,1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{A}\mathbf{R}_j^* - H(0)\mathbf{G}$ and $\mathbf{D}_{j,\theta} = \mathbf{A}\mathbf{R}_{j,\theta}^* + \mathbf{G}$

(iii) For each $j \in J_2^*$, $\mathcal{B}$ chooses $\mathbf{R}_j^* \longleftarrow \{-1,1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{A}\mathbf{R}_j^* - \mathbf{G}$

(iv) For each $j \in \mathcal{J}_2 \setminus J_2^*$, $\mathcal{B}$ chooses $\mathbf{R}_j^* \longleftarrow \{-1,1\}^{m\times m}$ and computes $\mathbf{B}_j = \mathbf{A}\mathbf{R}_j^*$

Finally, $\mathcal{B}$ sends the public key $\mathrm{PK} = \{\mathbf{A}, \{\mathbf{B}_i\}_{i\in\mathcal{J}}, \mathbf{u}, H, \mathrm{BT}\}$ to $\mathcal{A}$ and keeps $(\{\mathbf{R}_j^*\}_{j\in\mathcal{J}}, \{\mathbf{R}_{j,\theta}^*\}_{j\in\mathcal{J}_1,\theta\in BT_j}, \nu_u, \mathbf{v})$ secret.

Phases 1 and 2: when $\mathcal{B}$ receives a key generation query (id, S) from $\mathcal{A}$, where $S = \{\mathrm{att}_i\}_{i\in I}$, he outputs $\perp$ if $S_{\mathrm{id},\mathrm{RL}^*} = \{\mathrm{att}_i \in S | \mathrm{id} \notin \mathrm{RL}_i^*, i \in I\} \models (W^*, t^*)$. Otherwise, for each $i \in I$, $\mathcal{B}$ picks an unassigned leaf node $\nu_{\mathrm{id},i}$ from BT_i and stores id in that node. Let $S_{\mathrm{id},\mathrm{RL}^*} \cap W^* = \{\mathrm{att}_j\}_{j\in K}$, and we have $|K| < t^*$; thus, $|K \cup J_2^*| \le d$. Then, $\mathcal{B}$ chooses a set K' such that $K \cup J_2^* \subseteq K' \subseteq I \cup \mathcal{J}_2$ and $|K'| = d$.

For each $j \in K'$,

(i) If $j \in I$, choose $\mathbf{e}_j' \longleftarrow \mathcal{D}_{\mathbb{Z}^{2m}\sigma}$; let $\mathbf{E}_j' = (\mathbf{A}\|\mathbf{B}_j + H(\mathrm{att}_j)\mathbf{G})$; then, compute $\widehat{\mathbf{u}}_j' = \mathbf{E}_j'\cdot\mathbf{e}_j'$.

(1) If $j \in J_1^*$ and $\mathrm{id} \notin \mathrm{RL}_j^*$, there exists $\theta_j^* \in \mathrm{path}(\mathrm{BT}_j, \nu_{\mathrm{id},j}) \cap \mathrm{KUNodes}(\mathrm{BT}_j, \mathrm{RL}_j^*)$.

Choose $\mathbf{e}_{j,\theta_j^*}'' \longleftarrow \mathcal{D}_{\mathbb{Z}^{2m}\sigma}$, let $\mathbf{E}_{j,\theta_j^*}'' = (\mathbf{A}\|\mathbf{D}_{j,\theta_j^*})$, and compute $\widehat{\mathbf{u}}_j'' = \mathbf{E}_{j,\theta_j^*}''\cdot\mathbf{e}_{j,\theta_j^*}''$. For each $\theta \in \mathrm{path}(\mathrm{BT}_j, \nu_{\mathrm{id},j}) \setminus \{\theta_j^*\}$, let $\mathbf{E}_{j,\theta}'' = (\mathbf{A}\|\mathbf{D}_{j,\theta}) = (\mathbf{A}\|\mathbf{A}\mathbf{R}_{j,\theta}^* + \mathbf{G})$, and then sample $\mathbf{e}_{j,\theta}'' \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}_{j,\theta}^*, \widehat{\mathbf{u}}_j'', \mathbf{T}_G, \sigma)$ such that $\mathbf{E}_{j,\theta}''\cdot\mathbf{e}_{j,\theta}'' = \widehat{\mathbf{u}}_j''$.

(2) Else, pick $\widehat{\mathbf{u}}_j'' \longleftarrow \mathbb{Z}_q^n$. For $\theta \in \mathrm{path}(\mathrm{BT}_j, \nu_{\mathrm{id},j})$, let $\mathbf{E}_{j,\theta}'' = (\mathbf{A}\|\mathbf{D}_{j,\theta}) = (\mathbf{A}\|\mathbf{A}\mathbf{R}_{j,\theta}^* + \mathbf{G})$ and sample $\mathbf{e}_{j,\theta}'' \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}_{j,\theta}^*, \widehat{\mathbf{u}}_j'', \mathbf{T}_G, \sigma)$ such that $\mathbf{E}_{j,\theta}''\cdot\mathbf{e}_{j,\theta}'' = \widehat{\mathbf{u}}_j''$.

Then, $\mathcal{B}$ computes $\widehat{\mathbf{u}}_j = \widehat{\mathbf{u}}_j' + \widehat{\mathbf{u}}_j''$.

(i) If $j \in \mathcal{J}_2$, choose $\mathbf{e}_j \longleftarrow \mathcal{D}_{\mathbb{Z}^{2m}\sigma}$, let $\mathbf{E}_j = (\mathbf{A}\|\mathbf{B}_j + \mathbf{G})$, and compute $\widehat{\mathbf{u}}_j = \mathbf{E}_j\cdot\mathbf{e}_j$.

Let n degree d polynomials be $p_1(x), \ldots, p_n(x)$

such that $\mathbf{u} = (p_1(0), \ldots, p_n(0))$ and $\hat{\mathbf{u}}_j = (p_1(j), \ldots, p_n(j))$ for each $j \in K'$. Then, we can recover polynomials $p_1(x), \ldots, p_n(x) \in \mathbb{Z}_q[x]$ by the Lagrange interpolation formula. Compute $\hat{\mathbf{u}}_j = (p_1(j), \ldots, p_n(j))$ for each $j \in (I \cup \mathcal{I}_2) \setminus K'$.

For each $j \in I \setminus (K' \cap I)$,

(i) If $j \in J_1^*$ and $\mathrm{att}_j = \mathrm{att}_j^*$, we have $\mathrm{id} \in \mathrm{RL}_j^*$. Choose $\mathbf{e}_j' \longleftarrow \mathcal{D}_{\mathbb{Z}^{2m}\sigma}$, let $\mathbf{E}_j' = (\mathbf{A} \| \mathbf{B}_j + H(\mathrm{att}_j^*)\mathbf{G}) = (\mathbf{A} \| \mathbf{A}\mathbf{R}_j^*)$, and compute $\hat{\mathbf{u}}_j' = \mathbf{E}_j' \cdot \mathbf{e}_j'$ and $\hat{\mathbf{u}}_j'' = \hat{\mathbf{u}}_j - \hat{\mathbf{u}}_j'$. For each $\theta \in \mathrm{path}(\mathrm{BT}_j, \nu_{\mathrm{id},j})$, let $\mathbf{E}_{j,\theta}'' = (\mathbf{A} \| \mathbf{D}_{j,\theta}) = (\mathbf{A} \| \mathbf{A}\mathbf{R}_{j,\theta}^* + \mathbf{G})$, and $\mathcal{B}$ can sample $\mathbf{e}_{j,\theta}'' \sim \mathcal{D}_{\Lambda_q^{\hat{u}_j''}(E_{j,\theta}'')}\sigma$ by using the SampleRight algorithm.

(ii) If $j \in J_1^*$, $\mathrm{att}_j \neq \mathrm{att}_j^*$, and $\mathrm{id} \notin \mathrm{RL}_j^*$, there exists $\theta_j^* \in \mathrm{path}(\mathrm{BT}_j, \nu_{\mathrm{id},j}) \cap \mathrm{KUNodes}(\mathrm{BT}_j, \mathrm{RL}_j^*)$. Choose $\mathbf{e}_{j,\theta_j^*}'' \longleftarrow \mathcal{D}_{\mathbb{Z}^{2m}}\sigma$, let $\mathbf{E}_{j,\theta_j^*}'' = (\mathbf{A} \| \mathbf{D}_{j,\theta_j^*})$, and compute $\hat{\mathbf{u}}_j'' = \mathbf{E}_{j,\theta_j^*}'' \cdot \mathbf{e}_{j,\theta_j^*}''$. For $\theta \in \mathrm{path}(\mathrm{BT}_j, \nu_{\mathrm{id},j}) \setminus \{\theta_j^*\}$, let $\mathbf{E}_{j,\theta}'' = (\mathbf{A} \| \mathbf{D}_{j,\theta}) = (\mathbf{A} \| \mathbf{A}\mathbf{R}_{j,\theta}^* + \mathbf{G})$ and sample $\mathbf{e}_{j,\theta}'' \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}_{j,\theta}^*, \hat{\mathbf{u}}_j'', \mathbf{T}_G, \sigma)$ such that $\mathbf{E}_{j,\theta}'' \cdot \mathbf{e}_{j,\theta}'' = \hat{\mathbf{u}}_j''$. Then, compute $\hat{\mathbf{u}}_j' = \hat{\mathbf{u}}_j - \hat{\mathbf{u}}_j''$ and sample $\mathbf{e}_j' \sim \mathcal{D}_{\Lambda_q^{\hat{u}_j}(E_j')}\sigma$ by using the SampleRight algorithm, where $\mathbf{E}_j' = (\mathbf{A} \| \mathbf{B}_j + H(\mathrm{att}_j)\mathbf{G}) = (\mathbf{A} \| \mathbf{A}\mathbf{R}_j^* + (H(\mathrm{att}_j) - H(\mathrm{att}_j^*))\mathbf{G})$.

(iii) Otherwise, choose $\hat{\mathbf{u}}_j'' \longleftarrow \mathbb{Z}_q^n$ and compute $\hat{\mathbf{u}}_j' = \hat{\mathbf{u}}_j - \hat{\mathbf{u}}_j''$. For $\theta \in \mathrm{path}(\mathrm{BT}_j, \nu_{\mathrm{id},j})$, sample $\mathbf{e}_{j,\theta}'' \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}_{j,\theta}^*, \hat{\mathbf{u}}_j'', \mathbf{T}_G, \sigma)$ for $\mathbf{E}_{j,\theta}'' = (\mathbf{A} \| \mathbf{A}\mathbf{R}_{j,\theta}^* + \mathbf{G})$. Then, sample $\mathbf{e}_j' \sim \mathcal{D}_{\Lambda_q^{\hat{u}_j}(E_j')}\sigma$ by using the SampleRight algorithm, where $\mathbf{E}_j' = (\mathbf{A} \| \mathbf{A}\mathbf{R}_j^* + (H(\mathrm{att}_j) - H(\mathrm{att}_j^*))\mathbf{G})$ if $j \in J_1^*$ and $\mathbf{E}_j' = (\mathbf{A} \| \mathbf{A}\mathbf{R}_j^* + (H(\mathrm{att}_j) - H(0))\mathbf{G})$ if $j \notin J_1^*$.

For each $j \in \mathcal{I}_2 \setminus (K' \cap \mathcal{I}_2)$, let $\mathbf{E}_j = (\mathbf{A} \| \mathbf{B}_j + \mathbf{G}) = (\mathbf{A} \| \mathbf{A}\mathbf{R}_j^* + \mathbf{G})$ and sample $\mathbf{e}_j \longleftarrow \mathrm{SampleRight}\mathbf{A}, \mathbf{G}, \mathbf{R}_j^*, \hat{\mathbf{u}}_j, \mathbf{T}_\mathbf{G}, \sigma$.

Finally, $\mathcal{B}$ sends $\mathrm{sk}_{S,\mathrm{id}} = (\{\mathbf{e}_i'\}_{i \in I}, \{\mathbf{e}_{i,\theta}''\}_{i \in I, \theta \in \mathrm{path}(\mathrm{BT}_i, \nu_{\mathrm{id},I})}, \{\mathbf{e}_i\}_{i \in \mathcal{I}_2})$ to $\mathcal{A}$.

Challenge: when $\mathcal{A}$ submits two different messages $M_0, M_1 \in \{0,1\}$, $\mathcal{B}$ flips a random coin $b \in \{0,1\}$ and computes $c_0 = Dv_u + M_b\lfloor q/2 \rfloor$, $\mathbf{c} = D\mathbf{v}$. For each $j \in J_1^*$ and each $\theta \in \mathrm{KUNodesBT}_j\mathrm{RL}_j^*$, $\mathcal{B}$ computes $\mathbf{c}_j' = D(\mathbf{R}_j^*)^\top \mathbf{v}$ and $\mathbf{c}_{j,\theta}'' = D(\mathbf{R}_{j,\theta}^*)^\top \mathbf{v}$. For each $j \in J_2^*$, $\mathcal{B}$ computes $\mathbf{c}_j = D(\mathbf{R}_j^*)^\top \mathbf{v}$. Finally, $\mathcal{B}$ sends the ciphertext $C^* = (c_0, \mathbf{c}, \{\mathbf{c}_j'\}_{j \in J_1^*}, \{\mathbf{c}_{j,\theta}''\}_{j \in J_1^*, \theta \in \mathrm{KUNodesBT}_j\mathrm{RL}_j^*}, \{\mathbf{c}_j\}_{j \in J_2^*})$ to $\mathcal{A}$.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b. If $b' = b$, $\mathcal{B}$ outputs 1; else, $\mathcal{B}$ outputs 0.

Note that, by Lemma 3, the pair $(\mathbf{A}, \mathbf{u})$ is computationally indistinguishable from its distribution in the real attack. Applying Lemma 9, we know that $\{\mathbf{B}_i\}_{i \in \mathcal{I}}$ and $\{\mathbf{D}_{i,\theta}\}_{i \in \mathcal{I}_1, \theta \in \mathrm{BT}_i}$ are statistically close to uniform even given more information about $(\mathbf{R}_i^*)^\top \mathbf{x}$ and $(\mathbf{R}_{i,\theta}^*)^\top \mathbf{x}$, respectively. Hence, the distribution of the public key in the simulation is indistinguishable from that in the real attack, and $\mathcal{A}$ gains negligible information about $\{\mathbf{R}_i^*\}_{i \in \mathcal{I}}$ and $\{\mathbf{R}_{i,\theta}^*\}_{i \in \mathcal{I}_1, \theta \in \mathrm{BT}_i}$ from the public key. According to Lemmas 2, 6, and 7, the output distribution of the key generation simulation using the SampleRight algorithm is statistical to that in the real attack.

If $\mathcal{O}(\cdot) = \mathcal{O}_{s^*}$ for some $\mathbf{s}^*$, we claim that the challenge ciphertext C^* is a valid ciphertext for $\mathbf{s} = D\mathbf{s}^*$, $\{\mathbf{R}_i^*\}_{i \in J_1^* \cup J_2^*}$, and $\{\mathbf{R}_{i,\theta}^*\}_{i \in J_1^*, \theta \in \mathrm{KUNodesBT}_i\mathrm{RL}_i^*}$: note that, for each $j \in J_1^*$ and each $\theta \in \mathrm{KUNodes}(\mathrm{BT}_j, \mathrm{RL}_j^*)$, $\mathbf{c}_j' = D(\mathbf{R}_j^*)^\top(\mathbf{A}^\top \mathbf{s}^* + \mathbf{x}) = (\mathbf{A}\mathbf{R}_j^*)^\top(D\mathbf{s}^*) + D(\mathbf{R}_j^*)^\top \mathbf{x} = (\mathbf{B}_j + H(\mathrm{att}_j^*)\mathbf{G})^\top \mathbf{s} + D(\mathbf{R}_j^*)^\top \mathbf{x}$ and $\mathbf{c}_{j,\theta}'' = D(\mathbf{R}_{j,\theta}^*)^\top(\mathbf{A}^\top \mathbf{s}^* + \mathbf{x}) = (\mathbf{A}\mathbf{R}_{j,\theta}^*)^\top(D\mathbf{s}^*) + D(\mathbf{R}_{j,\theta}^*)^\top \mathbf{x} = \mathbf{D}_{j,\theta}^\top \cdot \mathbf{s} + D(\mathbf{R}_{j,\theta}^*)^\top \mathbf{x}$. For each $j \in J_2^*$, $\mathbf{c}_j = D(\mathbf{R}_j^*)^\top(\mathbf{A}^\top \mathbf{s}^* + \mathbf{x}) = (\mathbf{A}\mathbf{R}_j^*)^\top(D\mathbf{s}^*) + D(\mathbf{R}_j^*)^\top \mathbf{x} = (\mathbf{B}_j + \mathbf{G})^\top \mathbf{s} + D(\mathbf{R}_j^*)^\top \mathbf{x}$. Therefore, the ciphertext is the same as the view of $\mathcal{A}$ in the real attack. Hence, if $\mathcal{A}$ guesses right b with noticeable probability more than 1/2, then $\mathcal{B}$ can succeed in its game with the same probability. Else, if $\mathcal{O}(\cdot) = \mathcal{O}_\$$, by Theorem 3, c_0 and M_b are independent. Since M_b is uniformly distributed, the probability of $\mathcal{A}$ guesses right b is exactly 1/2. In a word, if $\mathcal{A}$ breaks the security of our DR-ABE, then $\mathcal{B}$ solves the underlying LWE problem. □

C. Reducing the Size of the Public Key

Our DR-ABE scheme with user-level (resp. attribute-level) revocation has a relatively large public key, and its dependence on the number of users N in the system is due to the fact that each node θ in BT (resp. each BT_i) is associated with a uniform random matrix $\mathbf{D}_\theta \in \mathbb{Z}_q^{n \times m}$ (resp. $\mathbf{D}_{i,\theta} \in \mathbb{Z}_q^{n \times m}$). In fact, the size of the public key can be reduced in the random oracle model in a way similar to [34]: let $\mathcal{H}$: $\{0,1\}^* \longrightarrow \mathbb{Z}_q^{n \times m}$ be a random oracle. For each node θ in BT (resp. each BT_i), we obtain uniformly random matrix $\mathbf{D}_\theta$ (resp. $\mathbf{D}_{i,\theta}$) as $\mathbf{D}_\theta := \mathcal{H}(\mathbf{A}, \{\mathbf{B}_j\}_{j \in \mathcal{I}}, \mathbf{u}, \theta)$ (resp. $\mathbf{D}_{i,\theta} := \mathcal{H}(\mathbf{A}, \{\mathbf{B}_j\}_{j \in \mathcal{I}}, \mathbf{u}, i, \theta)$). In the security proof, we first simulate the generation of $\mathbf{D}_\theta$ (resp. $\mathbf{D}_{i,\theta}$) as in the proof of Theorem 4 (resp. Theorem 5) and then program the random oracle $\mathcal{H}$ such that $\mathcal{H}(\mathbf{A}, \{\mathbf{B}_j\}_{j \in \mathcal{I}}, \mathbf{u}, \theta) = \mathbf{D}_\theta$ (resp. $\mathcal{H}(\mathbf{A}, \{\mathbf{B}_j\}_{j \in \mathcal{I}}, \mathbf{u}, i, \theta) = \mathbf{D}_{i,\theta}$).

D. Decryption Outsourcing

To make our schemes more applicable for the resource-limited end user, we modify our DR-ABE schemes to outsource most computational overhead of the end user to an honest-but-curious third party in the following manner: we add an extra dummy attribute dummy in the system. The

Setup algorithm chooses an extra matrix $\overline{\mathbf{B}} \longleftarrow \mathbb{Z}_q^{n\times m}$. To generate the private key for a user, the KGC splits the public vector $\mathbf{u}$ into $\overline{\mathbf{u}}, \hat{\mathbf{u}}$ such that $\mathbf{u} = \overline{\mathbf{u}} + \hat{\mathbf{u}}$, samples $\overline{\mathbf{e}} \longleftarrow \text{SampleLeft}(\mathbf{A}, \mathbf{B} + H(\text{dummy}), \mathbf{G}, \overline{\mathbf{u}}, \mathbf{T}_A, \sigma)$, replaces $\mathbf{u}$ with $\hat{\mathbf{u}}$ in the original Keygen algorithm to get $\text{sk}_{S,\text{id}}$, and finally returns $\text{sk}_{S,\text{id}}$ along with $\overline{\mathbf{e}}$ as the private key of the user. Moreover, we add an extra ciphertext corresponding with dummy, $\overline{\mathbf{c}} = (\mathbf{B} + H(\text{dummy})\mathbf{G})^\top \mathbf{s} + D\overline{\mathbf{R}}^\top \mathbf{x}$, into the output of the original Enc algorithm. In this case, the end user can give $\text{sk}_{S,\text{id}}$ to an untrusted third party to help decrypt the ciphertext except for $\overline{\mathbf{c}}$. The third party will return $\hat{\mathbf{u}}\mathbf{s} + \mathbf{e}$ and $\overline{\mathbf{c}}$ to the user, and the latter only needs to deal with $\overline{\mathbf{c}}$ using $\overline{\mathbf{e}}$ to recover the message.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Leixiao Cheng, Fei Meng, Xianmeng Meng, and Qixin Zhang are the main authors of the current paper. Specifically, Leixiao Cheng first brought the idea of AKC into this paper to construct revocable ABE resistant to quantum attacks and provided the main construction of two DR-ABE schemes and the formal security proof. She also wrote the initial draft of this paper. Fei Meng contributed to the construction detail and parameter analysis of those two schemes. Xianmeng Meng and Qixin Zhang contributed to carrying out additional analyses and revised the final version of this paper. All authors contributed to writing and revision and approved the final manuscript.

Acknowledgments

The authors were supported by the National Cryptography Development Fund (Grant no. MMJJ20180210) and the National Natural Science Foundation of China (Grant nos. 61832012 and 61672019).

References

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," *Quantum Information and Computation*, vol. 3, no. 4, pp. 317–344, 2003.

[3] W. S. Peter, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, Santa Fe, NM, USA, November 1994.

[4] D. Bernstein, *Introduction to Post-Quantum Cryptography*, Springer, Berlin, Germany, 2009.

[5] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 84–93, Baltimore, MD, USA, May 2005.

[6] E. Alkim, L. Ducas, T. Pöppelmann, and S. Peter, "Post-quantum key exchange—a new hope," in *Proceedings of the 25th USENIX Security Symposium, USENIX Security*, pp. 327–343, Austin, TX, USA, August 2016.

[7] J. Bos, C. Craig, D. Leo et al., "Frodo: take off the ring! practical, quantum-secure key exchange from LWE," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS'16*, pp. 1006–1018, New York, NY, USA, 2016.

[8] J. Ding, "A simple provably secure key exchange scheme based on the learning with errors problem," *IACR Cryptology ePrint Archive*, vol. 688, 2012.

[9] Z. Jin and Y. Zhao, "Optimal key consensus in presence of noise, CoRR, Abs/1611.06150," 2016.

[10] Z. Jin and Y. Zhao, "Optimal key consensus in presence of noise," *IACR Cryptology ePrint Archive*, vol. 1058, p. 2017, 2017.

[11] Z. Jin and Y. Zhao, "Generic and practical key establishment from lattice," in *Proceedings of the Applied Cryptography and Network Security—17th International Conference, ACNS 2019*, pp. 302–322, Bogota, CO, USA, June 2019.

[12] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," *Topics in Cryptology—CT-RSA 2011*, Springer, Berlin, Germany, pp. 319–339, 2011.

[13] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM*, vol. 60, no. 6, pp. 43:1–43:35, 2013.

[14] C. Peikert, "Lattice cryptography for the internet," in *Proceedings of the Post-Quantum Cryptography—6th International Workshop, PQCrypto 2014*, pp. 197–219, Waterloo, Canada, October 2014.

[15] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 34:1–34:40, 2009.

[16] Y. Zhao, Z. jin, B. Gong, and G. Sui, "Supporting documentation: KCL," National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[17] L. Cheng, Q. Wu and Y. Zhao, "Compact lossy and allutne trapdoor functions from lattice", *in Proceedings of the 13th International Conference on Information Security Practice and Experience (ISPEC-*, vol. 10701, pp. 279-296, Springer, Melbourne, Australia, December 2017.

[18] NIST CSRC and Cryptographic Techology Group, *Submission Requirements and Evaluation Criteria for Post-quantum Cryptography Standardization Process*, NIST, Gaithersburg, MD, USA, 2016.

[19] NIST, Post-quantum Cryptography, Round 1 Submissions, 2017, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[20] S. Amit and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Advances in Cryptology—EUROCRYPT 2005, Aarhus, Denmark, May 2005.

[21] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attribute-based encryption," in *Proceedings of the Computer Security—ESORICS 2016—21st European Symposium on*

Research in Computer Security, pp. 570–587, Heraklion, Greece, September 2016.

[22] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, "Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list," in *Proceedings of the 16th International Conference, ACNS 2018*, pp. 516–534, Applied Cryptography and Network Security, Leuven, Belgium, July 2018.

[23] B. Qin, Q. Zhao, Z. Dong, and H. Cui, "Server-aided revocable attribute-based encryption resilient to decryption key exposure," in *Proceedings of the 16th International Conference, CANS 2017*, pp. 504–514, Cryptology and Network Security, Hong Kong, China, November 2017.

[24] S. Amit, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Proceedings of the 32nd Annual Cryptology Conference*, pp. 199–217, Advances in Cryptology—CRYPTO 2012, Santa Barbara, CA, USA, August 2012.

[25] Y. Yang, X. Ding, H. Lu, Z. Wan, and J. Zhou, "Achieving revocable fine-grained cryptographic access control over cloud data," in *Proceedings of the Information Security, 16th International Conference, ISC 2013*, pp. 293–308, Dallas, TX, USA, November 2013.

[26] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010*, pp. 261–270, Beijing, China, April 2010.

[27] S. Jahid, P. Mittal, and N. Borisov, "Easier: encryption-based access control in social networks with efficient revocation," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2011*, pp. 411–415, Hong Kong, China, March 2011.

[28] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[29] Z. Xu and M. Keith, "Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012*, pp. 844–849, Liverpool, United Kingdom, June 2012.

[30] H. Lian, Q. Wang, and G. Wang, "Large universe ciphertext-policy attribute-based encryption with attribute level user revocation in cloud storage," *The International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 107–117, 2020.

[31] J. Chen, H. W. Lim, S. Ling, H. Wang, and K. Nguyen, "Revocable identity-based encryption from lattices," in *Proceedings of the Information Security and Privacy—17th Australasian Conference, ACISP 2012*, pp. 390–403, Wollongong Australia, July 2012.

[32] M. Pirretti, P. Traynor, P. D. McDaniel, and B. Waters, "Secure attribute-based systems," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pp. 99–112, Alexandria, VA, USA, November 2006.

[33] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Proceedings of the Third International Conference*, pp. 248–265, Pairing-Based Cryptography—Pairing 2009, Palo Alto, CA, USA, August 2009.

[34] S. Ling, K. Nguyen, H. Wang, and J. Zhang, "Revocable predicate encryption from lattices," in *Proceedings of the Provable Security—11th International Conference, ProvSec 2017*, pp. 305–326, Xi'an, China, October 2017.

[35] R. Ostrovsky, S. Amit, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, pp. 195–203, Alexandria, VA, USA, October 2007.

[36] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 553–572, Advances in Cryptology—EUROCRYPT 2010, French Riviera, Monaco, May 2010.

[37] R. Cramer and I. Damgård, "On the amortized complexity of zero-knowledge protocols," in *Proceedings of the 29th Annual International Cryptology Conference*, pp. 177–191, Advances in Cryptology—CRYPTO 2009, Santa Barbara, CA, USA, August 2009.

[38] M. Ajtai, "Generating hard instances of the short basis problem," in *Proceedings of the Automata, Languages and Programming, 26th International Colloquium, ICALP'99*, pp. 1–9, Prague, Czech Republic, July 1999.

[39] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, vol. 48, no. 3, pp. 535–553, 2011.

[40] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 700–718, Advances in Cryptology—EUROCRYPT 2012, Cambridge, UK, April 2012.

[41] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," *Journal of Cryptology*, vol. 25, no. 4, pp. 601–639, 2012.

[42] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," *IACR Cryptology ePrint Archive*, vol. 52, p. 2012, 2012.

[43] S. Katsumata, T. Matsuda, and A. Takayasu, "Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance," in *Proceedings of the 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography*, pp. 441–471, Public-Key Cryptography—PKC 2019, Beijing, China, April 2019.

[44] J. M. González Nieto, M. Manulis, and D. Sun, "Fully private revocable predicate encryption," in *Proceedings of the Information Security and Privacy—17th Australasian Conference, ACISP 2012*, pp. 350–363, Wollongong, Australia, July 2012.

[45] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: security model and construction," in *Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 216–234, Public-Key Cryptography—PKC 2013, Nara, Japan, February 2013.

[46] S. Wang, X. Zhang, and Y. Zhang, "Efficient revocable and grantable attribute-based encryption from lattices with fine-grained access control," *IET Information Security*, vol. 12, no. 2, pp. 141–149, 2018.

[47] Y. Kang, G. Wu, C. Dong, X. Fu, F. Li, and T. Wu, "Attribute based encryption with efficient revocation from lattices," *International Journal of Network Security*, vol. 22, no. 1, pp. 161–170, 2020.

[48] J. Zhang, Z. Zhang, and A. Ge, "Ciphertext policy attribute-based encryption from lattices," in *Proceedings of the 7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS'12*, pp. 16-17, Seoul, South Korea, May 2012.

[49] R. El Bansarkhani, "Supporting documentation: kindi," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[50] H. C. Jung, S. Park, J. Lee et al., "Supporting documentation: lizard," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[51] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, "Supporting documentation: saber," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[52] J. Ding, T. Takagi, X. Gao, and Y. Wang, "Supporting documentation: ding key exchange," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[53] O. Garcia-Morchon, Z. Zhang, S. Bhattacharya et al., "Supporting documentation: round2," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[54] X. Lu, Y. Liu, D. Jia, H. Xue, J. He, and Z. Zhang, "Supporting documentation: lac," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[55] M. Naehrig, E. Alkim, J. Bos et al., "Supporting documentation: frodokem," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[56] Le T. Phong, T. Hayashi, Y. Aono, and S. Moriai, "Supporting documentation: lotus," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[57] S. Peter, R. Avanzi, J. Bos et al., "Supporting documentation: kyber," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[58] M. Seo, J. Hwan Park, H. L. Dong, S. Kim, and S.-J. Lee, "Supporting documentation: emblem and r.emblem," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[59] N. P. Smart, M. R. Albrecht, Y. Lindell et al., "Supporting documentation: lima," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[60] R. Steinfeld, S. Amin, and K. Raymond, "Supporting documentation: titanium," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[61] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom functions and lattices," in *Proceedings of the International Conference on Theory and Applications of Cryptographic Techniques*, pp. 719–737, Cambgridge, UK, April 2012.

[62] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Lecture Notes in Computer Science*, Springer, 2010, pp. 21–23, Springer, Berlin, Germany, Advances in Cryptology—EUROCRYPT 2010.

[63] A. Langlois and D. Stehle, "Worst-case to average-case reductions for module lattices. cryptology ePrint archive," 2012, https://eprint.iacr.org/2012/090.

[64] M. Hamburg, "Supporting documentation: threebears," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[65] M.-J. Saarinen, "Supporting documentation: hila5," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[66] T. Pöppelmann, E. Alkim, R. Avanzi et al., "Supporting documentation: newhope," Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

[67] V. Goyal, O. Pandey, S. Amit, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pp. 89–98, Alexandria, VA, USA, October 2006.

[68] N. Attrapadung, B. Libert, and E. de Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proceedings of the Public Key Cryptography—PKC 2011—14th International Conference on Practice and Theory in Public Key Cryptography*, pp. 90–108, Taormina, Italy, March 2011.

[69] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Proceedings of the Public-Key Cryptography—PKC 2013—16th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 162–179, Nara, Japan, February 2013.

[70] J. Bethencourt, S. Amit, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P 2007)*, pp. 321–334, Oakland, CA, USA, May 2007.

[71] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography*, pp. 53–70, Public Key Cryptography—PKC 2011, Taormina, Italy, March 2011.

[72] F. Meng, L. Cheng, M. Wang, P. Voulgaris, and H. Wee, "ABDKS: attribute-based encryption with dynamic keyword search in fog computing," *Frontiers of Computer Science*.

[73] G. Craig, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 197–206, Victoria, Canada, May 2008.

[74] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pp. 333–342, Bethesda, MD, USA, May 2009.

[75] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee, "Fuzzy identity based encryption from lattices," *IACR Cryptology ePrint Archive*, vol. 414, p. 2011, 2011.