Hindawi
Security and Communication Networks
Volume 2020, Article ID 1460531, 13 pages
https://doi.org/10.1155/2020/1460531

Research Article

Server-Aided Revocable Attribute-Based Encryption from Lattices

Xingting Dong,[1] Yanhua Zhang,[2] Baocang Wang,[1] and Jiangshan Chen[3]

[1]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China*
[2]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China*
[3]*School of Mathematics and Statistics, Minnan Normal University, Zhangzhou 363000, China*

Correspondence should be addressed to Yanhua Zhang; yhzhang@zzuli.edu.cn

Received 16 October 2019; Accepted 18 December 2019; Published 12 February 2020

Academic Editor: Bruce M. Kapron

Attribute-based encryption (ABE) can support a fine-grained access control to encrypted data. When the user's secret-key is compromised, the ABE system has to revoke its decryption privileges to prevent the leakage of encrypted data. Although there are many constructions about revocable ABE from bilinear maps, the situation with lattice-based constructions is less satisfactory, and a few efforts were made to close this gap. In this work, we propose the first lattice-based server-aided revocable attribute-based encryption (SR-ABE) scheme and thus the first such construction that is believed to be quantum resistant. In the standard model, our scheme is proved to be secure based on the hardness of the Learning With Errors (LWE) problem.

1. Introduction

Attribute-based encryption (ABE) [1, 2], which was first introduced in 2006 as a generalization of identity-based encryption (IBE) [3, 4] and fuzzy identity-based encryption (FIBE) [1, 5], is such a notion for public-key encryption which is used to implement fine-grained access control. ABE system includes two types: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In KP-ABE, the key generation center (KGC) generates a master secret key (msk) and a master public key (mpk), and each user has a policy function f. The KGC computes and sends to the user a secret key sk_f corresponding to its policy function f. To encrypt a message μ, a sender selects some required attributes from the attribute set to form a subset att and generates the ciphertext $\mathrm{ct}_{\mathrm{att}}$ labeled with att. The recipient owning the policy function f can decrypt $\mathrm{ct}_{\mathrm{att}}$ by applying the secret key sk_f if and only if $f(\mathrm{att}) = 1$. The different between CP-ABE and KP-ABE is that in CP-ABE, each user has its own attribute subset att and a ciphertext is corresponding to a policy function f.

Several important results are proposed to realize ABE in the last few years. These schemes have several types. One of these can be implemented to predicates computable by Boolean formulas [2, 6–11] (which are limited to log-depth computations). Another of these has made some important progress [12–16], which can apply to sophisticated circuits. In 2013, based on Learning With Errors (LWE) problem, Gorbunov et al. proposed a KP-ABE scheme [16], which is called GVW13, where its predicate can be arbitrary polynomial-size circuits. It is one of the important candidates for Boolean circuit ABE.

When the users in ABE system changed, for example some users leave the system or their secret keys are leaked, these users' secret keys should be revoked from the system. In other words, although these users have legal secret keys sk_f, they cannot decrypt ciphertext after leaving the system. So for an ABE system with a large number of users, an efficient revocable mechanism is very necessary and important.

In the beginning, revocation mechanism is introduced into IBE. To address the user revocation mechanism, in 2008, Boldyreva et al. [17] proposed the first revocation scheme by combining the complete subtree method [18] with FIBE. After the work of Boldyreva et al. [17], a lot of studies [19, 20] have been put forward. In 2013, in response to many realistic threats and attack scenarios, a new security notion unique to the revocation scheme called decryption

key exposure resistance (DKER) was proposed by Seo and Emura [20–23]. Since then, DKER has quickly become an important security requirement for RIBE and many follow-up RIBE schemes with DKER [24–28] were proposed. In order to improve the efficiency of revocation, in 2015, Qin et al. [29] proposed an interesting solution, called server-aided revocable IBE (SR-IBE). In their scheme, a publicly accessible server with powerful computational capabilities, which can be untrusted in the sense that it does not possess any secret information, is used to outsource most of the users workload.

The revocable ABE scheme appears later. In 2009, Attrapadung and Imai [30] put forward two revocable methods. One is direct revocation which is that the sender should specify the revocation list while encrypting, and the other is indirect revocation. In the indirect revocation scheme, in order to achieve the key revocation mechanism, each user's secret key cannot be allowed to decrypt ciphertexts alone. To complete the decryption, the KGC broadcasts key update through a public channel for every time period. The key update is useless for revoked users, but nonrevoked users will be allowed to combine their secret keys with the key update to derive a decryption key, which can finally decrypt ciphertexts. And they proposed the first hybrid revocable ABE scheme.

In 2010, Yu et al. [31] proposed a indirect revocable ABE; however, policy function only supports logical AND. In 2012, Amit et al. [32] provided a more generic way to achieve indirect revocation in ABE schemes. In order to alleviate the workload of users, in 2013, Yang et al. [33] proposed a direct revocable ABE scheme by delegating part of the users decryption capability to a semitrusted server, however, which results in an increase in traffic over the secret channel. To mitigate user's workload and the traffic of the secret channel, in 2016, Cui et al. [34] proposed a scheme called server-aided revocable ABE (SR-ABE) based on the large universe CP-ABE scheme. If the server in their scheme was colluded with an adversary, however, the SR-ABE may be not DKER. To solve this problem, based on [34], in 2017, Qin et al. [35] proposed a SR-ABE with DKER. About direct revocable, in 2018, Liu et al. [36] proposed an efficient revocable CP-ABE scheme by embedding the revocation list into ciphertext. And they have a shorter revocation list.

These RIBE and RABE schemes operate in the bilinear pairing setting; however, the system has narrowed in the race to protect sensitive electronic information from the threat of quantum computers, which one day could render these constructions obsolete. Up to now, known quantum algorithms have no obvious advantages (beyond polynomial speedup) over classical ones in solving problems in lattice such as shortest vector problem (SVP), closest vector problem (CVP), short integer solution (SIS), and LWE. Lattice-based cryptography is considered as an ideal candidate for postquantum cryptography (PQC), and possesses several noticeable advantages over conventional number-theoretic cryptography (i.e., based on integer factoring or discrete logarithm problems), such as conjectured resistance against quantum computers, faster arithmetic operations, and provable security under the worst-case hardness assumptions. And among the PQC schemes submitted to NIST, lattice-based schemes are the most.

In 2012, Chen et al. [37] proposed the first RIBE scheme from lattices without DKER. In 2017, Takayasu and Watanabe [38] proposed a variant of [37] and partially solved the problem of achieving RIBE with DKER. In 2019, Katsumata et al. [39] completely solved the problem of achieving RIBE with DKER by proposing the first lattice-based RIBE scheme with DKER secure under the LWE assumption.

But the progress in constructing revocable ABE schemes from lattices is slow. In 2018, Ling et al. [40] proposed a server-aided revocable Predicate Encryption (SR-PE) from LWE. This scheme employs the Predicate Encryption (PE) scheme of Agrawal et al. [41] and the complete subtree method of Naor et al. [18] as the two main ingredients, and plus some additional techniques. In the security proof of the SR-PE, however, since the LWE secret vector in the original PE scheme is unknown, an unreasonable challenge ciphertext is constructed, leading to an invalid proof.

1.1. Our Contributions. In order to solve the security of revocable ABE against quantum attack, we propose the first SR-ABE from LWE which is indirect revocable and satisfies efficient and secure user revocation in lattices. In order to mitigate the burden of users, all the work caused by the revocation will be delegated to a powerful untrusted server. The powerful server is similar to cloud computing, with a large number of computing resources and storage resources, which can ensure the correctness of the calculation, but cannot guarantee the security of the data. In order to achieve the key revocation mechanism, in our scheme each user's secret key cannot be allowed to decrypt ciphertexts alone. To complete the decryption, KGC should bind the user's identity and corresponding circuit when generating the public key, and bind the period time when generating update key. When the user's identity is not revoked and its circuit matches the attribute subset of the ciphertext, the server can generate a transformation key from KGC to convert the ciphertext into a partially decrypted ciphertext bound only with the identity. In this way, only the secret key of the identity can be used to decrypt. The framework of our SR-ABE scheme is depicted in Figure 1.

In the scheme, there are four types of participants: a KGC, a powerful untrusted server, data sender and data recipient, among which the KGC and the server are the service of the system, the data sender and the data recipient are the client of the system. The server is opened to anyone, including the adversary. In our scheme, a user's policy function is a boolean circuit $\mathscr{C}_{\mathrm{id}}$ with its identity id.

According to the system parameter, the KGC generates an msk and an mpk and broadcasts the mpk to all users. By using its msk and a user's identity id, KGC can generate a secret key $\mathrm{sk}_{\mathrm{id}}$ which is sent to the user. When a data μ needs to be sent, a data sender specifies an attribute subset att for the data, encrypts it over the att and a time period t by using the mpk, and sends the ciphertext $\mathrm{ct}_{t,\mathrm{att}}$ to the server. And all data users can see the ciphertext on the server. If needing to

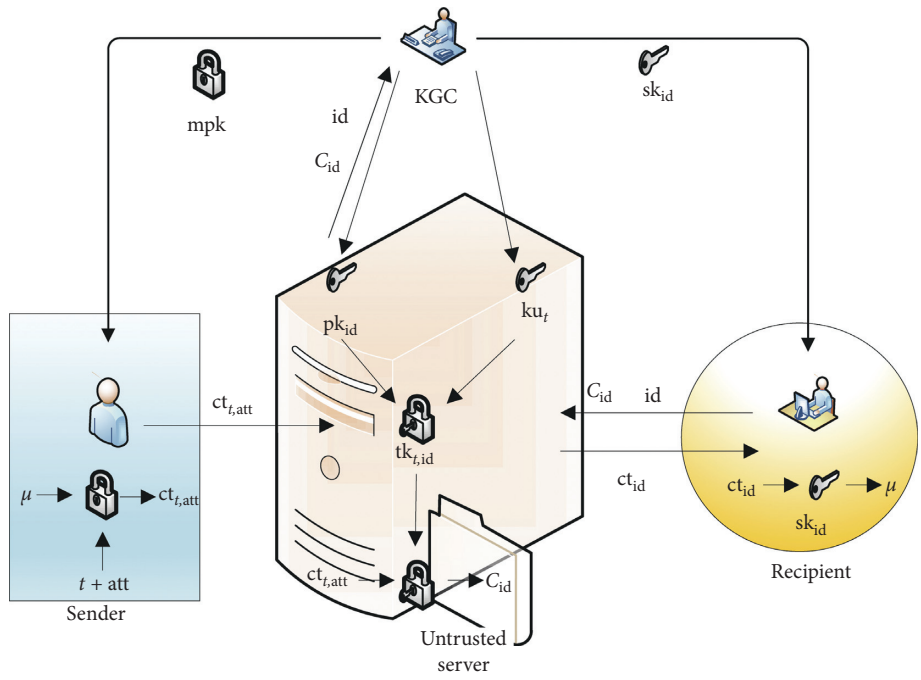

Figure 1: Framework of server-aided revocable attribute-based encryption.

decrypt a ciphertext $ct_{t,att}$, a data recipient with identity id forwards its identity id and corresponding circuit $\mathscr{C}_{id}$ to the server and points out the ciphertext that it wants the server to decrypt. And the server sends $\mathscr{C}_{id}$ and id to the KGC. If $\mathscr{C}_{id}$ corresponds to the attribute subset of the ciphertext, KGC can generate a public key pk_{id} binding identity id for this user and send pk_{id} to server. And also the KGC can generate a key update ku_t for nonrevoked user in a time period t and send it to server. If the recipient's identity id is not in revocation list RL, then the server is able to generate a transformation key $tk_{t,id}$ by using the key update ku_t and public key pk_{id}. With the key $tk_{t,id}$, the server can get partially decrypted ciphertext ct_{id} of binding identity id and send it to the data recipient. Finally the recipient can decrypt ct_{id} completely by using its secret key.

The public key pk_{id} is bound to an identity id and corresponding circuit $\mathscr{C}_{id}$, which results in the transformation key $tk_{t,id}$ binding the identity and circuit as well. If the $\mathscr{C}_{id}(att) = 1$, then the server can use $tk_{t,id}$ to separate the attribute subset att and time t from the ciphertext and bind the identity id to generate the partial decrypted ciphertext ct_{id}. The partial decrypted ciphertext can only be decrypted by the secret key sk_{id} corresponding the identity which the recipient sends to the server.

According to the security model of GVW13 ABE [16], we define selective security model for our SR-ABE from LWE which takes into account the possible realistic threats in selective security model and formalizes all attack strategies of an adversary against the SR-ABE scheme. The selective security model is that the adversary needs to give the challenge attributes att^* and challenge time period t^* before seeing the master public key mpk. There are two attack strategies, one is that when the adversary can access the secret key sk_{id^*} of a user with identity id^* whose circuit $\mathscr{C}_{id^*}$ matches att^* within t^*, the identity id^* should be in revocation list before t^*, and the other is that if this identity id^* has not been revoked in t^*, the adversary can not query the secret key sk_{id^*} corresponding to this id^* with $\mathscr{C}_{id^*}(att^*) = 1$.

In short, our contributions in this paper can be summed up in the following three points:

(i) We formally define the SR-ABE model from lattices that support Boolean circuit of any arbitrary polynominal size. We give the definition of the correctness and security of SR-ABE from LWE.

(ii) We propose a concrete SR-ABE construction from lattice for this model based on the KP-ABE constructed by Gorbunov et al. [16].

(iii) We give a strict proof of security for our scheme, based on the hardness of Learning With Errors problem and prove that our SR-ABE scheme is selective security if the GVW13 is selective security.

1.2. Organization. In the forthcoming sections, we first introduce the nations and definitions relevant to this paper in Section 2. We construct the first lattice-based SR-ABE scheme in Section 3 and analyze the correctness and security and compare our scheme with previous revocable schemes in Section 4. We conclude the paper in Section 5.

2. Preliminaries

2.1. Notation. Bold capital letters (e.g., $\mathbf{A}$) denote matrices, bold lowercase letters (e.g., $\mathbf{a}$) denote vectors. The probabilistic polynomial time algorithm is denoted by PPT. $[\ell]$ denotes the set of $\{1, \ldots, \ell\}$ where $\ell \in \mathbb{Z}$. For a vector $\mathbf{a}$, $\|\mathbf{a}\|$

denotes its Euclidean norm. A nonnegative function $\mathrm{negl}(n)$ is *negligible* if, for every polynomial $p(n)$, it holds that $\mathrm{negl}(n) \leq 1/p(n)$ for all sufficiently large $n > 0$.

2.2. Server-Aided Revocable Attribute-Based Encryption. In order to support a class of boolean circuits $\mathscr{C}$ we add several parameters to conventional SR-ABE where ℓ denotes the length of attributes and $d_{\max}$ denotes the depth of a boolean circuit $\mathscr{C}$.

2.2.1. Syntax of SR-ABE. A SR-ABE scheme consists of ten following polynomial-time algorithms:

(1) $\mathrm{System}(1^\lambda, 1^\ell, d_{\max}) \longrightarrow (\mathrm{pp})$: the KGC takes a security parameter λ, an attribute length ℓ, and a circuit depth $d_{\max}$ as input and outputs the system parameter pp.

(2) $\mathrm{Setup}(\mathrm{pp}) \longrightarrow (\mathrm{mpk}, \mathrm{msk}, \mathrm{RL}, \mathrm{st})$: the KGC takes the parameter pp as input, and outputs a master public key mpk, a master secret key msk, a revocation list RL, and a state st.

(3) $\mathrm{GenSK}(\mathrm{msk}, \mathrm{id}) \longrightarrow (\mathrm{sk}_{\mathrm{id}})$: the KGC takes msk, identity id as input; outputs the user secret key $\mathrm{sk}_{\mathrm{id}}$; and sends it to the user with the identity id.

(4) $\mathrm{Encrypt}(\mathrm{mpk}, t, \mu, \mathrm{att}) \longrightarrow (\mathrm{ct}_{t,\mathrm{att}})$: the sender takes mpk, a time $t \in \mathscr{T}$, a message $\mu \in \mathscr{M}$ and an attribute subset att as input; outputs the ciphertext $\mathrm{ct}_{t,\mathrm{att}}$; and sends it to the server.

(5) $\mathrm{GenPK}(\mathrm{msk}, \mathrm{id}, \mathscr{C}_{\mathrm{id}}, \mathrm{st}) \longrightarrow (\mathrm{pk}_{\mathrm{id}}, \mathrm{st}')$: the KGC takes msk, an identity id, a circuit $\mathscr{C}_{\mathrm{id}}$ corresponding to id, and a state st as input; outputs the public key $\mathrm{pk}_{\mathrm{id}}$ with identity id and updates the state to st'; and sends $\mathrm{pk}_{\mathrm{id}}$ to the server.

(6) $\mathrm{KeyUp}(\mathrm{msk}, t, \mathrm{RL}, \mathrm{st}) \longrightarrow (\mathrm{ku}_t, \mathrm{st}')$: the KGC takes msk, a time $t \in \mathscr{T}$, a revocation list RL, and a state st as input; outputs a key update ku_t and updates the state to st'; and sends ku_t to the server.

(7) $\mathrm{TranKG}(\mathrm{pk}_{\mathrm{id}}, \mathrm{ku}_t) \longrightarrow (\mathrm{tk}_{t,\mathrm{id}}/\perp)$: the server takes the public key $\mathrm{pk}_{\mathrm{id}}$ with identity id and a key update ku_t as input, and if $\mathrm{id} \notin \mathrm{RL}$, and outputs a transform key $\mathrm{tk}_{t,\mathrm{id}}$ for a user with identity id, else outputs $\perp$.

(8) $\mathrm{Transform}(\mathrm{ct}_{t,\mathrm{att}}, \mathrm{tk}_{t,\mathrm{id}}) \longrightarrow (\mathrm{ct}_{\mathrm{id}}/\perp)$: the server takes the ciphertext $\mathrm{ct}_{t,\mathrm{att}}$ and a transform key $\mathrm{tk}_{t,\mathrm{id}}$ as input and, if the circuit $\mathscr{C}_{\mathrm{id}}$ corresponding to $\mathrm{pk}_{\mathrm{id}}$ in $\mathrm{tk}_{t,\mathrm{id}}$ satisfies $\mathscr{C}_{\mathrm{id}}(\mathrm{att}) = 1$, outputs a partially decrypted ciphertext $\mathrm{ct}_{\mathrm{id}}$ with identity id and sends it to the recipient, else outputs $\perp$.

(9) $\mathrm{Dec}(\mathrm{ct}_{\mathrm{id}}, \mathrm{sk}_{\mathrm{id}}) \longrightarrow (\mu')$: the recipient with identity id takes the partially decrypted ciphertext $\mathrm{ct}_{\mathrm{id}}$ and its secret key $\mathrm{sk}_{\mathrm{id}}$ as input and outputs the message μ'.

(10) $\mathrm{Revoke}(\{\mathrm{id}\}_{\mathrm{id}\in U}, t, \mathrm{RL}, \mathrm{st}) \longrightarrow (\mathrm{RL}, \mathrm{st}')$: the KGC takes an identity set $\{\mathrm{id}\}_{\mathrm{id}\in U}$, time t, the revocation list RL, and the current state st, and outputs a new RL and updates the state to st'.

Definition 1 (correctness of SR-ABE). The correctness of SR-ABE requires that for all security parameter λ, the circuit depth $d_{\max}$, the attribute length ℓ, all message $\mu \in \mathscr{M}$, all $t \in \mathscr{T}$, and $(\mathrm{msk}, \mathrm{mpk}, \mathrm{RL}, \mathrm{st}) \longleftarrow \mathrm{Setup}(\mathrm{pp})$, if the user with identity $\mathrm{id} \notin \mathrm{RL}$ by time t and $\mathscr{C}_{\mathrm{id}}(\mathrm{att}) = 1$ and all parties follow the scheme's algorithms, then for all ciphertexts $\mathrm{ct}_{t,\mathrm{att}} \longleftarrow \mathrm{Encrypt}(\mathrm{mpk}, t, \mu, \mathrm{att})$, there exists $\mathrm{sk}_{\mathrm{id}} \longleftarrow \mathrm{GenSK}(\mathrm{msk}, \mathrm{id})$, for $\mathrm{tk}_{t,\mathrm{id}} \longleftarrow \mathrm{TranKG}(\mathrm{pk}_{\mathrm{id}}, \mathrm{ku}_t)$ and $\mathrm{ct}_{\mathrm{id}} \longleftarrow \mathrm{Transform}(\mathrm{ct}_{t,\mathrm{att}}, \mathrm{tk}_{t,\mathrm{id}})$, such that it has $\mathrm{Dec}(\mathrm{ct}_{\mathrm{id}}, \mathrm{sk}_{\mathrm{id}}) = \mu$ where $\mathrm{pk}_{\mathrm{id}} \longleftarrow \mathrm{GenPK}(\mathrm{msk}, \mathrm{id}, \mathscr{C}_{\mathrm{id}}, \mathrm{st})$, and $\mathrm{ku}_t \longleftarrow \mathrm{KeyUp}(\mathrm{msk}, t, \mathrm{RL}, \mathrm{st})$.

Chen et al. [37] formalized and defined the selective-revocable-identity security revocable IBE from lattices. Qin et al. [35] defined the IND-CPA security model for SR-ABE from bilinear pairings. In this subsection, we give the definition of selective attribute security server-aided revocable attribute-based encryption from lattices.

2.2.2. Selective Security Game. An adversary $\mathscr{A}$ and a challenger $\mathscr{S}$ play the following game.

Initial $\mathscr{A}$ first gives the challenge attributes att^* and time t^*, and some information state it wants to preserve.

Setup $\mathscr{S}$ runs the $\mathrm{Setup}(\cdot)$, generates the msk, mpk, RL and st and sends mpk, RL and st to $\mathscr{A}$.

Query $\mathscr{A}$ can adaptively make a polynomial number of following queries to $\mathscr{S}$.

$\mathrm{GenSK}(\cdot)$: on input identity id and circuit $\mathscr{C}_{\mathrm{id}}$ corresponding to id, return a secret key $\mathrm{sk}_{\mathrm{id}}$.
$\mathrm{GenPK}(\cdot)$: on input identity id, circuit $\mathscr{C}_{\mathrm{id}}$ corresponding to id and a state st, return $\mathrm{pk}_{\mathrm{id}}$.
$\mathrm{KeyUp}(\cdot)$: on input time t, revocation list RL and state st, return ku_t.
$\mathrm{TranKG}(\cdot)$: on input ku_t and $\mathrm{pk}_{\mathrm{id}}$ with identity id, if $\mathrm{id} \notin \mathrm{RL}$ return $\mathrm{tk}_{t,\mathrm{id}}$, and else return $\perp$.
$\mathrm{Transform}(\cdot)$: on input the ciphertext $\mathrm{ct}_{t,\mathrm{att}}$ and circuit $\mathscr{C}_{\mathrm{id}}$ with identity id and $\mathrm{tk}_{t,\mathrm{id}}$, if $\mathscr{C}_{\mathrm{id}}(\mathrm{att}) = 1$ outputs partially decrypted ciphertext $\mathrm{ct}_{\mathrm{att}}$, else outputs $\perp$.
$\mathrm{Revoke}(\cdot)$: on input identity id, time t and state st, return updated revocation list RL.

The following restrictions must always hold:

If id^* with $\mathscr{C}_{\mathrm{id}^*}(\mathrm{att}^*) = 1$ has been queried to $\mathrm{GenSK}(\cdot)$ at t^*, the $\mathrm{Revoke}(\cdot)$ must be queried on (id^*, t) for any $t \leq t^*$.
If id^* with $\mathscr{C}_{\mathrm{id}^*}(\mathrm{att}^*) = 1$ is not revoked at t^*, $(\mathrm{id}^*, \mathscr{C}_{\mathrm{id}})$ should not be queried to the $\mathrm{GenSK}(\cdot)$.

Challenge $\mathscr{A}$ outputs two equal length message $\mu_0, \mu_1 \in \mathscr{M}$ and sends them to $\mathscr{S}$. $\mathscr{S}$ randomly chooses a bit $\beta \in \{0, 1\}$ and sends $\mathrm{Encrypt}(\mathrm{mpk}, t^*, \mu_\beta, \mathrm{att}^*)$ to $\mathscr{A}$.

Guess $\mathscr{A}$ can continue to make a polynomial numbers of queries as in Query phase and outputs a bit β'. $\mathscr{A}$ will win if $\beta' = \beta$.

Definition 2 (selective security). The advantage of $\mathcal{A}$ is defined as the quantity:

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{SR-ABE}}\left(1^{\lambda}, 1^{\ell}, d_{\max}\right) := \Pr\left[\beta = \beta'\right] - \frac{1}{2}. \quad (1)$$

The scheme SR-ABE is called to be selective security, if the advantage of adversary $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{SR-ABE}}(1^{\lambda}, 1^{\ell}, d_{\max})$ is negligible in $\lambda, \ell, d_{\max}$ for an efficient $\mathcal{A}$.

2.3. Background on Lattices

Definition 3 (lattices). Let q, n, m be positive integers; for a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}_q^m: \mathbf{Ax} = 0 \mod q\}$ denotes an certain family of integer lattices which was introduced by Ajtai [42]. More generally, for $\mathbf{u} \in \mathbb{Z}_q^n$, $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ denotes the coset $\{\mathbf{x} \in \mathbb{Z}_q^m: \mathbf{Ax} = \mathbf{u} \mod q\}$.

Definition 4 (discrete Gaussians). For a vector $\mathbf{c} \in \mathbb{R}^m$, a parameter $s > 0$ and an integer lattice Λ, define $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi(\|\mathbf{x} - \mathbf{c}\|^2/s^2))$ and $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda}\rho_{s,\mathbf{c}}(\mathbf{x})$. The discrete Gaussians distribution over lattice Λ with center vector $\mathbf{c}$ and a parameter s is $\forall \mathbf{x} \in \Lambda, \mathcal{D}_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/\rho_{s,\mathbf{c}}(\Lambda)$. We will simplify to use notations $\mathcal{D}_{\Lambda,s}$ when $\mathbf{c} = 0$.

Definition 5 (learning with errors (LWE)). LWE was introduced by Regev [43]. For positive integers n, m, a prime integer q, and a discrete Gaussians distribution $\chi = \mathcal{D}_{\mathbb{Z},s}$. The decisional $\mathrm{LWE}_{n,q,\chi}$ problem is to distinguish the following two distributions: a uniform distribution pair $(\mathbf{A}, \mathbf{b})$ where $(\mathbf{A}, \mathbf{b}) \longleftarrow \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, and the other distribution pair $(\mathbf{A}, \mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e})$ where $(\mathbf{A}, \mathbf{s}) \longleftarrow \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$ and $\mathbf{e} \longleftarrow \chi^m$.

Some efficient sampling algorithms which find some short vectors from specific lattice were introduced by Agrawal et al. [44] and Micciancio and Peikert [45]. We recall these sampling algorithms.

Lemma 1. *For positive integers $n \geq 1, q \geq 2$ and efficiently large $m = O(n \log q)$, There are polynomial time algorithms with the properties below:*

(1) $\mathrm{TrapGen}(n, m, q) \longrightarrow \mathbf{A}, \mathbf{T}_{\mathbf{A}}$: an efficient randomized algorithm [45–47], outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^{\perp}(\mathbf{A})$ such that the distribute of $\mathbf{A}$ is close to uniform and $\|\widetilde{\mathbf{T}_{\mathbf{A}}}\| \leq O(\sqrt{m \log q})$, $\|\mathbf{T}_{\mathbf{A}}\| \leq O(m \log q)$ where $\widetilde{\mathbf{T}_{\mathbf{A}}}$ denotes Gram–Schmidt orthogonalization of $\mathbf{T}_{\mathbf{A}}$.

(2) $\mathrm{SampleLeft}(\mathbf{A}, \mathbf{M}, \mathbf{T}_{\mathbf{A}}, \mathbf{u}, s)$: inputting $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a trapdoor $\mathbf{T}_{\mathbf{A}}$ of $\Lambda_q^{\perp}(\mathbf{A})$, a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times m}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a sufficiently large Gaussian parameter $s \geq \|\widetilde{\mathbf{T}_{\mathbf{A}}}\| \cdot \omega(\sqrt{\log 2m})$, it outputs a vector $\mathbf{z} \in \mathbb{Z}^{2m}$ with a distribute statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}([\mathbf{A}|\mathbf{M}]),s}$.

(3) $\mathrm{SampleRight}(\mathbf{A}, \mathbf{R}, \mathbf{G}, \mathbf{T}_{\mathbf{G}}, \mathbf{u}, s)$: inputting $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$, a trapdoor $\mathbf{T}_{\mathbf{G}}$ of $\Lambda_q^{\perp}(\mathbf{G})$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a sufficiently large Gaussian parameter $s \geq \|\mathbf{T}_{\mathbf{G}}\| \cdot \|\mathbf{R}\| \cdot \omega(\sqrt{\log m})$, it outputs a vector $\mathbf{z} \in \mathbb{Z}^{2m}$ with a distribute statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}([\mathbf{A}|\mathbf{AR}+\mathbf{G}]),s}$.

2.4. Two-To-One Recoding Scheme.

In this subsection, we will introduce the Two-to-One Recoding (TOR) scheme simply presented by Gorbunov et al. based on LWE in [16]. And its idea is introduced in [44, 46, 48, 49].

Lemma 2. *Assuming the Decisional $\mathrm{LWE}_{n,q,\chi}$, there is a TOR.*

(1) $\mathrm{Params}(1^{\lambda}, d_{max})$: on input parameter λ and $d_{\max}$, output (m, n, q).

(2) $\mathrm{Keygen}(m, n, q)$: on input parameter m, n, q, run $\mathrm{TrapGen}(n, m, q)$ and get a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{T}$ of $\Lambda_q^{\perp}(\mathbf{A})$. And output $pk = \mathbf{A}, sk = \mathbf{T}$.

(3) $\mathrm{Encode}(pk, \mathbf{s} \in \mathbb{Z}_q^n)$: output the encoding $\psi = \mathbf{A}^T\mathbf{s} + \mathbf{e} \in \mathbb{Z}^m$, where $\mathbf{e} \longleftarrow \mathcal{D}_{\mathbb{Z}^m,s}$. ψ is called an encoding of $\mathbf{s}$, and $\mathbf{e}$ is called error vector.

(4) $\mathrm{ReKeygen}(pk_0, pk_1, sk_b, pk_{tgt})$: let $pk_b = \mathbf{A}_b$, $sk_b = \mathbf{T}_b$, $pk_{tgt} = \mathbf{A}_{tgt}$ for $b \in \{0, 1\}$. Compute $\mathbf{R} \in \mathbb{Z}^{2m\times m}$

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \end{bmatrix}, \mathbf{R}_i \in \mathbb{Z}^{m\times m}, \quad i = 0, 1, \quad (2)$$

where $\mathbf{R}_1 \longleftarrow \mathcal{D}_{\mathbb{Z}^{m\times m},s}$, and $\mathbf{R}_0 \longleftarrow \mathrm{SamplePre}(\mathbf{A}_0, \mathbf{T}_0, \mathbf{U}, s)$, where $\mathbf{U} = \mathbf{A}_{tgt} - \mathbf{A}_1\mathbf{R}_1$. Output $rk_{0,1}^{tgt} = \mathbf{R}$.

(5) $\mathrm{SimReKeyGen}(pk_0, pk_1)$: let $pk_0 = \mathbf{A}_0$, $pk_1 = \mathbf{A}_1$, and sample a matrix $\mathbf{R} \longleftarrow (\mathcal{D}_{\mathbb{Z}^{2m\times m},s})$. Define $\mathbf{A}_{tgt} := [\mathbf{A}_0 \mid \mathbf{A}_1]\mathbf{R} \in \mathbb{Z}_q^{n\times m}$ and output the pair $(pk_{tgt} = \mathbf{A}_{tgt}, rk_{0,1}^{tgt} = \mathbf{R})$.

(6) $\mathrm{Recode}(rk_{0,1}^{tgt}, \psi_0, \psi_1)$: let $rk_{0,1}^{tgt} = \mathbf{R}$ and compute

$$\psi_{\mathrm{tgt}} = \mathbf{R}^T \begin{bmatrix} \psi_0 \\ \psi_1 \end{bmatrix} \in \mathbb{Z}_q^m, \quad (3)$$

where $\psi_0 = \mathrm{Encode}(\mathbf{A}_0, \mathbf{s})$, $\psi_1 = \mathrm{Encode}(\mathbf{A}_1, \mathbf{s})$ for same $\mathbf{s} \in \mathbb{Z}^n$. It is clear that $\psi_{tgt} = \mathrm{Encode}(\mathbf{A}_{tgt}, \mathbf{s})$ for same $\mathbf{s} \in \mathbb{Z}^n$ as long as the error-tolerance is large enough. Output ψ_{tgt}.

The ABE scheme needs a one-time symmetric encryption scheme (E, D) which is in the following.

Lemma 3. *Let $\mu \in \{0, 1\}^m$ denote the plaintext, γ denote corresponding ciphertext, ψ and $\psi' \in \mathbb{Z}_q^m$, then*

(i) $E(\psi, \mu)$: compute the ciphertext $\gamma = \psi + [q/2]\mu \pmod q$. And output γ.

(ii) $D(\psi', \gamma)$: let $\psi' = (\psi_0', \ldots, \psi_{m-1}') \in \mathbb{Z}_q^m$ and a ciphertext $\gamma = (\gamma_0, \ldots, \gamma_{m-1}) \in \mathbb{Z}_q^m$, compute

$$\mu' = \left(\mathrm{Round}(\gamma_0 - \psi_0'), \mathrm{Round}(\gamma_1 - \psi_1'), \ldots, \mathrm{Round} \cdot (\gamma_{m-1} - \psi_{m-1})\right), \quad (4)$$

where

$$\text{Round}(x) = \begin{cases} 0, & \text{if } |x \bmod q| < q/4, \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

Output μ'.

2.5. Full-Rank Different Map

Definition 6 (full-rank different map [37]). Let q be a prime and n a positive integer. A function $H\colon \mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n\times n}$ is a full-rank different map, if for all different vectors $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, the matrix $H(\mathbf{u}) - H(\mathbf{v}) \in \mathbb{Z}_q^{n\times n}$ is full rank and H is computable in polynomial time in $n\log q$.

2.6. Complete Subtree Method. Like previous revocable schemes, our scheme also needs to use the complete subtree method which was proposed by Naor et al. [18]. In the method, there is a complete binary BT with at least N leaf nodes, where N is the maximum number of users in the system and each leaf node of BT is corresponding to a user. With this binary tree BT, a KUNode algorithm is used to compute the minimal set of nodes for which key update needs to be published so that only the nonrevoked users in this tree at a time period t are able to decrypt the ciphertexts.

KUNode(BT, RL, t) takes the binary tree BT, a revocation list RL, and a time period t as input and does the following:

(1) $X, Y \longleftarrow \varnothing$;

(2) $\forall (x_i, t_i) \in \text{RL}$, if $t_i \le t$, then add $\text{Path}(x_i)$ to X;

(3) $\forall y \in X$, if $y_l \notin X$, then add y_l to Y, if $y_r \notin X$, then add y_r to Y, where y_l is left child of y and y_r is right child of y;

(4) if $Y = \varnothing$, then add root to Y;

(5) Return Y.

The set Y is the smallest subset of nodes that contains ancestors of all the leaf nodes corresponding to nonrevoked users. In [18], it proves that the set Y generated by KUNodes(BT, RL, t) has a size at most $O(R\log N/R)$, where R is the number of users in RL.

3. SR-ABE from Lattices

3.1. GVW'13 ABE Scheme. In this subsection, we will briefly describe GVW13 ABE scheme [16], which will be used as the building block for our SR-ABE.

There are three key parameters in GVW13 ABE Scheme, which are security parameter λ, attribute length ℓ, and circuit depth $d_{\max}$, respectively. The master public key is $\left\{\left\{\mathbf{A}_{i,j}\right\}_{i\in[\ell], j\in\{0,1\}}, \mathbf{A}_{\text{out}}\right\}$ and master secret key is $\left\{\mathbf{T}_{i,j}\right\}_{i\in[\ell], j\in\{0,1\}}$, where $(\mathbf{A}_{i,j}, \mathbf{T}_{i,j}) \longleftarrow \text{KeyGen}(\cdot)$ for $i \in [\ell]$, $j \in \{0,1\}$. The generation of the secret key for a user with a circuit $\mathscr{C}$ is complex. First of all, the KGC assigns the $(\mathbf{A}_{i,b}, \mathbf{T}_{i,b}) \longleftarrow \text{Kengen}(\cdot)$ to every output $b \in \{0,1\}$ of the i-th gate of the circuit $\mathscr{C}$ for $i \in \{\ell+1, \ldots, |\mathscr{C}|-1\}$. When $i = |\mathscr{C}|$, the last gate is assigned $\mathbf{A}_{\text{out}}$ only when the output of the gate is 1. Then, according to every gate $\mathscr{C}_i$ of the circuit $\mathscr{C}$, the conversation keys are generated by $rk_{b,c}^i \longleftarrow \text{ReKeyGen}(\mathbf{A}_{i-2,b}, \mathbf{A}_{i-1,c}, \mathbf{T}_{i-2,b}, \mathbf{A}_{i,a})$ where $a = \mathscr{C}_i(b,c)$ and $b, c \in \{0,1\}$. Finally, these conversation keys are combined as user's secret key, and distributed to the user. If a message μ needs to be sent, according to the $\text{att} = \{a_1, a_2, \ldots, a_\ell\} \in \{0,1\}^\ell$, a sender selects $\left\{\mathbf{A}_{i,a_i}\right\}_{i\in[\ell]}$ to encrypt it and gets the ciphertext $\{\text{att}, \{\text{Encode}(\mathbf{A}_{i,a_i}, \mathbf{u})\}_{i\in[\ell]}, E(\text{Encode}(\mathbf{A}_{\text{out}}, \mathbf{u}), \mu)\}$ where $\mathbf{u} \longleftarrow \mathbb{Z}_q^n$. When a recipient with the circuit $\mathscr{C}$ wants to decrypt the ciphertext, if $\mathscr{C}(\text{att}) = 1$, then it can use secret key to get the code of $\mathbf{A}_{\text{out}}$ according to the code of $\left\{\mathbf{A}_{i,a_i}\right\}_{i\in[\ell]}$ and can easily get the message μ; else, it can do nothing.

In the selective security model, the adversary announces a challenge attribute set att^* before the challenger gives it public master key. According to [16], the GVW13 scheme is selectively secure.

3.2. Our SR-ABE Scheme. In this subsection, we give a concrete construction of our scheme.

3.2.1. System$(1^\lambda, 1^\ell, d_{max})$. On input the λ, ℓ, and $d_{\max}$, the KGC does the following:

(1) Set $n = O(\lambda)$, $m = O(n\log q)$, the modulus $q = O(n^2 d_{\max})^{d_{\max}} n$, and Gaussian parameter $s = O(\sqrt{n\log q})$. Error distribution is $\chi = \mathscr{D}_{\mathbb{Z},\sqrt{n}}$. $N = \text{poly}(\lambda)$ is the maximal number of users the system can support. An efficient full-rank different map $H\colon \mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n\times n}$.

(2) Let the identify space be $\mathscr{I} \subseteq \mathbb{Z}_q^n$, the time space be $\mathscr{T} \subseteq \mathbb{Z}_q^n$, the message space be $\mathscr{M} \subseteq \{0,1\}^m$, and the attribute space be $\mathbb{A} \subseteq \{0,1\}^\ell$.

(3) Output $pp = (\ell, n, m, q, s, N, \chi, \mathscr{I}, \mathscr{T}, \mathscr{M}, H, \mathbb{A})$.

3.2.2. Setup(pp). On input pp, the KGC does the following:

(1) For $b \in \{0,1\}, i = 1, \ldots, \ell$, run $\text{Keygen}(m, n, q)$, and output $(\mathbf{A}, \mathbf{T}_\mathbf{A})$, $(\mathbf{B}, \mathbf{T}_\mathbf{B})$, and $\left\{(\mathbf{B}_{i,b}, \mathbf{T}_{i,b}^\mathbf{B})\right\}_{i\in[\ell], b\in\{0,1\}}$. Output

$$\begin{aligned} \text{pk}_1 &= (\mathbf{A}, \mathbf{B}), \\ \text{sk}_1 &= (\mathbf{T}_\mathbf{A}, \mathbf{T}_\mathbf{B}), \\ \text{pk}_2 &= \begin{pmatrix} \mathbf{B}_{1,0} & \mathbf{B}_{2,0} & \cdots & \mathbf{B}_{\ell,0} \\ \mathbf{B}_{1,1} & \mathbf{B}_{2,1} & \cdots & \mathbf{B}_{\ell,1} \end{pmatrix}, \\ \text{sk}_2 &= \begin{pmatrix} \mathbf{T}_{1,0}^\mathbf{B} & \mathbf{T}_{2,0}^\mathbf{B} & \cdots & \mathbf{T}_{\ell,0}^\mathbf{B} \\ \mathbf{T}_{1,1}^\mathbf{B} & \mathbf{T}_{2,1}^\mathbf{B} & \cdots & \mathbf{T}_{\ell,1}^\mathbf{B} \end{pmatrix}. \end{aligned} \quad (6)$$

(2) Choose randomly $\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}, \mathbf{D}, \mathbf{G} \longleftarrow \mathbb{Z}_q^{n\times m}$ and let $\text{msk} = (\text{sk}_1, \text{sk}_2, \mathbf{A}_1, \mathbf{B}_1)$ and $\text{mpk} = (\text{pk}_1, \text{pk}_2, \mathbf{C}, \mathbf{D}, \mathbf{G})$.

(3) Initialize the revocation list $\text{RL} = \varnothing$. Obtain a binary tree BT with at least N leaf nodes and set the state st = BT.

(4) Output (mpk, msk, RL, st).

3.2.3. GenSK (msk, id). On input msk, an identity id $\in \mathcal{I}$, the KGC does the following:

(1) If the $\mathbf{F}_{\mathrm{id}}$ corresponding to id is undefined, set $\mathbf{F}_{\mathrm{id}} = \mathbf{A}_1 + H(\mathrm{id})\mathbf{G}$, sample $\mathbf{R}_{\mathrm{id}} \longleftarrow \mathrm{SampleLeft}(\mathbf{A}, \mathbf{F}_{\mathrm{id}}, \mathbf{T}_{\mathbf{A}}, \mathbf{D}, s)$ and note that $[\mathbf{A} \mid \mathbf{F}_{\mathrm{id}}]\mathbf{R}_{\mathrm{id}} = \mathbf{D}$.

(2) Output $\mathrm{sk}_{\mathrm{id}} = \mathbf{R}_{\mathrm{id}}$.

3.2.4. Encrypt (mpk, t, μ, att). On input mpk, a time $\mathbf{t} \in \mathcal{T}$, and a message $\mu \in \mathcal{M}$, the sender selects an attribute subset $\mathrm{att} = (a_1, a_2, \ldots, a_\ell) \in \mathbb{A}$ and does the following:

(1) Set $\mathbf{C}_{\mathbf{t}} = \mathbf{C} + H(\mathbf{t})\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ and sample $\mathbf{u} \longleftarrow \mathbb{Z}_q^n$.

(2) Output $\mathrm{ct}_{t,\mathrm{att}} = (\mathrm{att}, \gamma, \{\psi_i\}_{i\in[\ell]}, \psi, \xi, \varphi)$, where

$$\begin{cases} \gamma = E(\mathrm{Encode}(\mathbf{D}, \mathbf{u}), \mu), \\ \psi_i = \mathrm{Encode}\left(\mathbf{B}_{i,a_i}, \mathbf{u}\right), \quad i \in [\ell], \\ \psi = \mathrm{Encode}(\mathbf{B}, \mathbf{u}), \\ \xi = \mathrm{Encode}(\mathbf{C}_{\mathbf{t}}, \mathbf{u}), \\ \varphi = \mathrm{Encode}(\mathbf{A}, \mathbf{u}). \end{cases} \tag{7}$$

3.2.5. GenPK $(msk, id, \mathscr{C}_{id}, st)$. On input msk, an identity id, a circuit $\mathscr{C}_{\mathrm{id}}$, and state st, the KGC does the following:

(1) For every leaf node θ from BT, store the corresponding identity id in this node. If the $\mathbf{B}_{\mathrm{id}}$ corresponding to id is undefined, set $\mathbf{B}_{\mathrm{id}} = \mathbf{B}_1 + H(\mathrm{id})\mathbf{G}$.

(2) After getting the circuit $\mathscr{C}_{\mathrm{id}}$ from server with identity id, for $i < |\mathscr{C}_{\mathrm{id}}| - \ell$ or $b = 0$, run $Keygen(pp)$ and get $(\mathbf{B}_{\ell+i,b}, \mathbf{T}^{\mathbf{B}}_{\ell+i,b})$. Set $\mathbf{B}_{|\mathscr{C}_{\mathrm{id}}|,1} = \mathbf{B}_{\mathrm{id}}$. For the gate $x_{\ell+i} = \mathscr{C}_{\mathrm{id}_i}(x_{u_i}, x_{v_i})$, $(b', b'') \in \{0,1\}^2$, $i = 1, \ldots, |\mathscr{C}_{\mathrm{id}}| - \ell$, there is $\mathbf{R}^{(u_i,v_i,\ell+i)}_{\mathbf{B}(b',b'',\mathscr{C}_{id_i}(b',b''))} \longleftarrow \mathrm{ReKeygen}(\mathbf{B}_{u_i,b'}, \mathbf{B}_{v_i,b''}, \mathbf{T}^{\mathbf{B}}_{u_i,b'}, \mathbf{B}_{\ell+i,\mathscr{C}_{id_i}(b',b'')})$. Let $s_{\mathrm{id}} = \left\{\mathbf{R}_{\mathbf{B}(b',b'',\mathscr{C}_{\mathrm{id}_i}(b',b''))}\right.$ $\left.(u_i, v_i, \ell+i), (b', b'') \in \{0,1\}^2, i = 1, \ldots, |\mathscr{C}_{\mathrm{id}}| - \ell\right\}$.

(3) For each node $x \in \mathrm{Path}(\theta)$, if its $\mathbf{U}_x$ is undefined, choose $\mathbf{U}_x \longleftarrow \mathbb{Z}_q^{n\times m}$ and store it on x. If the $\mathbf{F}_{\mathrm{id}}$ corresponding to id is undefined, set $\mathbf{F}_{\mathrm{id}} = \mathbf{A}_1 + H(\mathrm{id})\mathbf{G}$. Sample $\mathbf{Z}_{1,x} \longleftarrow \mathrm{SampleLeft}(\mathbf{B}, \mathbf{B}_{\mathrm{id}}, \mathbf{T}_{\mathbf{B}}, \mathbf{F}_{\mathrm{id}} - \mathbf{U}_x, s)$, and such that $[\mathbf{B} \mid \mathbf{B}_{\mathrm{id}}]\mathbf{Z}_{1,x} = \mathbf{F}_{\mathrm{id}} - \mathbf{U}_x$ where $\mathbf{Z}_{1,x} \in \mathscr{D}_{\Lambda^{\mathbf{B}_{\mathrm{id}}-\mathbf{U}_x}([\mathbf{B}\mid\mathbf{B}_{\mathrm{id}}]),s}$. And update the state to st'

(4) Output $\mathrm{pk}_{\mathrm{id}} = (s_{\mathrm{id}}, \left\{(x, \mathbf{Z}_{1,x})\right\}_{x\in\mathrm{Path}(\mathrm{id})})$ and the updated st'.

3.2.6. KeyUp (msk, t, RL, st). On input msk, a time $\mathbf{t} \in \mathcal{T}$, a revocation list RL, and the state st, the KGC dose the following:

(1) Set $\mathbf{C}_{\mathbf{t}} = \mathbf{C} + H(\mathbf{t})\mathbf{G} \in \mathbb{Z}_q^{n\times m}$.

(2) For all $x \in \mathrm{KUNodes}(\mathrm{BT}, \mathrm{RL}, \mathbf{t})$, fetch $\mathbf{U}_x$ from node x, and sample $\mathbf{Z}_{2,x} \longleftarrow \mathrm{SampleLeft}(\mathbf{B}, \mathbf{C}_{\mathbf{t}}, \mathbf{T}_{\mathbf{B}}, \mathbf{U}_x, s)$. Note that $\mathbf{Z}_{2,x} \in \mathscr{D}_{\Lambda^{\mathbf{U}_x}([\mathbf{B}\mid\mathbf{C}_{\mathbf{t}}]),s}$ and $[\mathbf{B} \mid \mathbf{C}_{\mathbf{t}}]\mathbf{Z}_{2,x} = \mathbf{U}_x$ (the corresponding $\mathbf{U}_x$ is predefined in GenPK and always exists). And update the state to st'.

(3) Output $\mathrm{ku}_t = \left\{(x, \mathbf{Z}_{2,x})\right\}_{x\in\mathrm{KUNodes}(\mathrm{BT},\mathrm{RL},t)}$ and the updated st'.

3.2.7. TranKG (pk_{id}, ku_t). On input $\mathrm{pk}_{\mathrm{id}}$ and ku_t, the server generates a transformation key $\mathrm{tk}_{t,\mathrm{id}}$ for every id not lying the revocation list RL as the following:

(1) Parse $\mathrm{pk}_{\mathrm{id}} = (s_{\mathrm{id}}, \left\{(x, \mathbf{Z}_{1,x})\right\}_{x\in I})$ and $\mathrm{ku}_t = \left\{(x, \mathbf{Z}_{2,x})\right\}_{x\in J}$ for some set of nodes I, J.

(2) If $I \cap J = \varnothing$, output $\perp$.

(3) Else choose $x = I \cap J$ and output $\mathrm{tk}_{t,\mathrm{id}} = (s_{\mathrm{id}}, \mathbf{Z}_{1,x}, \mathbf{Z}_{2,x})$. Note that $[\mathbf{B} \mid \mathbf{B}_{\mathrm{id}}]\mathbf{Z}_{1,x} + [\mathbf{B} \mid \mathbf{C}_{\mathbf{t}}]\mathbf{Z}_{2,x} = \mathbf{F}_{\mathrm{id}}$.

3.2.8. Transform $(ct_{t,att}, tk_{t,id})$. Receiving $\mathrm{tk}_{t,\mathrm{id}} = (s_{\mathrm{id}}, \mathbf{Z}_{1,x}, \mathbf{Z}_{2,x})$, the server does the following:

(1) If $\mathscr{C}_{\mathrm{id}}(\mathrm{att}) = 1$, use the key s_{id} to obtain $\psi_{\mathscr{C}_{\mathrm{id}}} = \mathrm{Encode}(\mathbf{B}_{\mathrm{id}}, \mathbf{u})$, else output $\perp$.

(2) Compute $\psi_{\mathrm{id}} = \mathbf{Z}_{1,x}^T \begin{bmatrix} \psi \\ \psi_{\mathscr{C}_{\mathrm{id}}} \end{bmatrix} + \mathbf{Z}_{2,x}^T \begin{bmatrix} \psi \\ \xi \end{bmatrix}$.

(3) Output $\mathrm{ct}_{\mathrm{id}} = (\mathrm{id}, \gamma, \varphi, \psi_{\mathrm{id}})$

The server sends $\mathrm{ct}_{\mathrm{att}}$ to the recipient with identify id.

3.2.9. Dec (ct_{id}, sk_{id}). On input $\mathrm{ct}_{\mathrm{id}}$ and secret key $\mathrm{sk}_{\mathrm{id}}$, the recipient can obtain $\mu' \longleftarrow D\left(\mathbf{R}_{\mathrm{id}}^T \begin{bmatrix} \varphi \\ \psi_{\mathrm{id}} \end{bmatrix}, \gamma\right)$ by using the secret key $\mathrm{sk}_{\mathrm{id}}$.

3.2.10. Revoke $(\{id\}_{id\in U}, t, RL, st)$. Taking an identity set $\{\mathrm{id}\}_{\mathrm{id}\in U}$ where U is a set of revoked users, time $\mathbf{t}$, the revocation list RL, and the current state st as input, the KGC adds $\mathrm{id} \in U$ to RL, updates the state to st', and outputs RL.

4. Correctness and Security Analysis

4.1. Correctness. When a recipient with $\mathrm{id} \notin \mathrm{RL}$ sends the circuit $\mathscr{C}_{\mathrm{id}}$ with $\mathscr{C}_{\mathrm{id}}(\mathrm{att}) = 1$ to server and wants to decrypt the ciphertext $\mathrm{ct}_{t,\mathrm{att}} = (\mathrm{att}, \gamma, \{\psi_i\}_{i\in[\ell]}, \psi, \xi, \varphi)$, the server and recipient perform as following.

(1) After accepting the circuit $\mathscr{C}_{\mathrm{id}}$ from the recipient, the server can send the $\mathscr{C}_{\mathrm{id}}$ to KGC and get $\mathrm{pk}_{\mathrm{id}} = (s_{\mathrm{id}}, \left\{(x, \mathbf{Z}_{1,x})\right\}_{x\in\mathrm{Path}(\mathrm{id})})$. And using the $\mathrm{ku}_t = \left\{(x, \mathbf{Z}_{2,x})\right\}_{x\in\mathrm{KUNodes}(\mathrm{BT},\mathrm{RL},t)}$, the server can get $\mathrm{tk}_{t,\mathrm{id}} = (s_{\mathrm{id}}, \mathbf{Z}_{1,x}, \mathbf{Z}_{2,x})$. By using the secret key s_{id} in $\mathrm{tk}_{t,\mathrm{id}}$ and $\{\psi_i\}_{i\in[\ell]}$ in $\mathrm{ct}_{t,\mathrm{att}}$, the server computes $\psi_{\mathscr{C}_{\mathrm{id}}} = \mathrm{Encode}(\mathbf{B}_{\mathrm{id}}, \mathbf{u})$, i.e., $\psi_{\mathscr{C}_{\mathrm{id}}} = \mathbf{B}_{\mathrm{id}}^T\mathbf{u} + \mathbf{e}_1$ where $\|e_1\| \le 2(n^3 \log^2 q)^{d_{\max}}$.

(2) Compute

$$\begin{aligned}\psi_{\mathrm{id}} &= \mathbf{Z}_{1,x}^T\begin{bmatrix}\psi\\ \psi_{\mathscr{C}_{\mathrm{id}}}\end{bmatrix} + \mathbf{Z}_{2,x}^T\begin{bmatrix}\psi\\ \xi\end{bmatrix}\\ &= \mathbf{Z}_{1,x}^T\begin{bmatrix}\mathbf{B}^T\mathbf{u}+\mathbf{e}_2\\ \mathbf{B}_{\mathrm{id}}^T\mathbf{u}+\mathbf{e}_1\end{bmatrix} + \mathbf{Z}_{2,x}^T\begin{bmatrix}\mathbf{B}^T\mathbf{u}+\mathbf{e}_2\\ \mathbf{C}_{\mathbf{t}}^T\mathbf{u}+\mathbf{e}_3\end{bmatrix}\\ &= \mathbf{Z}_{1,x}^T\left[\mathbf{B}\,\middle|\,\mathbf{B}_{\mathrm{id}}\right]^T\mathbf{u} + \mathbf{Z}_{1,x}^T\left[\mathbf{B}\,\middle|\,\mathbf{C}_t\right]^T\mathbf{u} + \mathbf{Z}_{1,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_1\end{bmatrix} \qquad (8)\\ &\quad + \mathbf{Z}_{2,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_3\end{bmatrix}\\ &= \mathbf{F}_{\mathrm{id}}^T\mathbf{u} + \mathbf{Z}_{1,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_1\end{bmatrix} + \mathbf{Z}_{2,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_3\end{bmatrix},\end{aligned}$$

where $\mathbf{e}_2, \mathbf{e}_3 \in \chi^m$.

Because of $\|\mathbf{e}_i\| = O(n)$, $\|\mathbf{Z}_{i,x}\| \le s\sqrt{m}$ for $i \in \{2,3\}$, then we have $\left\|\mathbf{Z}_{1,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_1\end{bmatrix} + \mathbf{Z}_{2,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_3\end{bmatrix}\right\| \le 4(n^3\log^2 q)^{d_{\max}}$ and then $\psi_{\mathrm{id}} = \mathrm{Encode}(\mathbf{F}_{\mathrm{id}}, \mathbf{u}) = \mathbf{F}_{\mathrm{id}}^T\mathbf{u} + \mathbf{e}_4$ where $\mathbf{e}_4 = \mathbf{Z}_{1,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_1\end{bmatrix} + \mathbf{Z}_{2,x}^T\begin{bmatrix}\mathbf{e}_2\\ \mathbf{e}_3\end{bmatrix}$. The server hands $\mathrm{ct}_{\mathrm{id}} = (\mathrm{id}, \gamma, \varphi, \psi_{\mathrm{id}})$ to recipient.

Receiving $\mathrm{ct}_{\mathrm{id}}$, the recipient uses the secret key $\mathrm{sk}_{\mathrm{id}}$ and computes

$$\begin{aligned}\gamma - \mathbf{R}_{\mathrm{id}}^T\begin{bmatrix}\varphi\\ \psi_{\mathrm{id}}\end{bmatrix} &= \mathbf{D}^T\mathbf{u} + \mathbf{e}_5 + \mu\left\lfloor\frac{q}{2}\right\rfloor - \mathbf{R}_{\mathrm{id}}^T\begin{bmatrix}\mathbf{A}^T\mathbf{u}+\mathbf{e}_6\\ \mathbf{F}_{\mathrm{id}}^T\mathbf{u}+\mathbf{e}_4\end{bmatrix}\\ &= \mathbf{D}^T\mathbf{u} + \mathbf{e}_5 + \mu\left\lfloor\frac{q}{2}\right\rfloor - \mathbf{R}_{\mathrm{id}}^T\left[\mathbf{A}\,\middle|\,\mathbf{F}_{\mathrm{id}}\right]^T\mathbf{u} - \mathbf{R}_{\mathrm{id}}^T\begin{bmatrix}\mathbf{e}_6\\ \mathbf{e}_4\end{bmatrix}\\ &= \mu\left\lfloor\frac{q}{2}\right\rfloor + \mathbf{e}_5 - \mathbf{R}_{\mathrm{id}}^T\begin{bmatrix}\mathbf{e}_6\\ \mathbf{e}_4\end{bmatrix}.\end{aligned} \qquad (9)$$

If $\left\|\mathbf{e}_5 - \mathbf{R}_{\mathrm{id}}^T\begin{bmatrix}\mathbf{e}_6\\ \mathbf{e}_4\end{bmatrix}\right\| \le 8(n^3\log^2 q)^{d_{\max}} < (q/4)$, then running decryption algorithm $D\left(\mathbf{R}_{\mathrm{id}}^T\begin{bmatrix}\varphi\\ \psi_{\mathrm{id}}\end{bmatrix}, \gamma\right)$, the recipient will obtain the message μ.

4.2. Security

Theorem 1. *Our SR-ABE scheme with attribute length ℓ is selective security defined in Definition 2 if the GVW13 scheme with attribute length $\ell + 2$ is selective security.*

Proof. If there exists a PPT adversary $\mathscr{A}$ against selective security of the SR-ABE scheme with attribute length ℓ, then we can construct a PPT adversary $\mathscr{B}$ against selective security of the GVW13 scheme with attribute length $\ell + 2$. The security of GVW13 scheme is based on LWE, so is our scheme.

Before proving this theorem, let us summarize our ideas of proof. In the GVW13 scheme with attribute length +2, we set $\mathbf{A} = \mathbf{B}_{l+1,0}$, $\mathbf{B} = \mathbf{B}_{l+2,0}$. And then our scheme's challenge ciphertext with $\mathrm{att}^* = \{a_1^*, a_2^*, \ldots, a_\ell^*\}$ can be regarded as a transformation of the challenge ciphertext of GVW13 scheme under attribute $\mathrm{att}^{*\prime} = \{a_1^*, a_2^*, \ldots, a_\ell^*, 0, 0\}$. Let us start with our proof.

In the GVW13 selective security model, after generating the system parameters λ, ℓ, and $d_{\max}$, the challenger $\mathscr{S}$ runs the System, gets pp, and gives the pp to $\mathscr{B}$. $\mathscr{B}$ hands it over to $\mathscr{A}$. Then $\mathscr{A}$ chooses a challenge attribute $\mathrm{att}^* \in \mathbb{A}$, a challenge time $\mathbf{t}^* \in \mathscr{T}$, and a revocation list RL^* and gives them to $\mathscr{B}$. Then $\mathscr{B}$ gives $\{\mathrm{att}^*, 0, 0\}$ to $\mathscr{S}$. Now, we consider two type of adversaries as follows:

Type I: it is assumed that every identity id^* whose circuit $\mathscr{C}_{\mathrm{id}^*}$ satisfies that $\mathscr{C}_{\mathrm{id}^*}(\mathrm{att}^*) = 1$ must be included in RL^*. In this case, $\mathscr{A}$ is allowed to issue a query to oracle $\mathrm{GenSK}(\cdot)$ on id^*.

Type II: it is assumed that there is an $\mathrm{id}^* \notin \mathrm{RL}^*$ whose circuit $\mathscr{C}_{\mathrm{id}^*}$ satisfies that $\mathscr{C}_{\mathrm{id}^*}(\mathrm{att}^*) = 1$. In this case, id^* is not revoked at t^* and $\mathscr{A}$ never issues a query to oracle $\mathrm{GenSK}(\cdot)$ on $(\mathrm{id}^*, \mathscr{C}_{\mathrm{id}^*})$.

The following steps are taken after $\mathscr{B}$ receiving the public key:

$$\mathrm{mpk}_{\mathrm{GVW13}} = \begin{pmatrix}\mathbf{B}_{1,0} & \mathbf{B}_{2,0} & \cdots & \mathbf{B}_{l,0} & \mathbf{B}_{l+1,0} & \mathbf{B}_{l+2,0}\\ \mathbf{B}_{1,1} & \mathbf{B}_{2,1} & \cdots & \mathbf{B}_{l,1} & \mathbf{B}_{l+1,1} & \mathbf{B}_{l+2,1} & \mathbf{B}_{\mathrm{out}}\end{pmatrix}, \qquad (10)$$

from $\mathscr{S}$.

(1) Generate $(\mathbf{G}, \mathbf{T}_{\mathbf{G}}) \longleftarrow \mathrm{TrapGen}(n, q, m)$, and set $\mathbf{A} = \mathbf{B}_{l+1,0}$, $\mathbf{B} = \mathbf{B}_{l+2,0}$.

(2) Sample $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3 \longleftarrow \{-1, 1\}^{m\times m}$. Choose an efficient full-rank different map $H: \mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n\times n}$. Choose an identity id^* with $\mathscr{C}_{\mathrm{id}^*}(\mathrm{att}^*) = 1$ and set $\mathbf{A}_1 = \mathbf{A}\mathbf{R}_1 - H(\mathrm{id}^*)\mathbf{G}$, $\mathbf{B} = \mathbf{B}\mathbf{R}_2 - H(\mathrm{id}^*)\mathbf{G}$, and $\mathbf{C} = \mathbf{B}\mathbf{R}_3 - H(\mathbf{t}^*)\mathbf{G}$.

(3′) *Type I adversary*: $\mathscr{B}$ can set revocation list RL^* and then sample $\mathbf{R}_{\mathrm{id}^*} = \begin{bmatrix}\mathbf{R}'\\ \mathbf{R}''\end{bmatrix} \longleftarrow \mathscr{D}_{\mathbb{Z}^{2m},s}$. Set $\mathbf{D} = [\mathbf{A}\,|\,\mathbf{A}\mathbf{R}_1]\mathbf{R}_{\mathrm{id}^*}$, and then let $\mathrm{mpk} = ((\mathbf{A}, \mathbf{B}), (\{\mathbf{B}_{i,b}\}_{i\in[l],b\in\{0,1\}}), \mathbf{C}, \mathbf{D}, \mathbf{G})$ and send mpk to the adversary $\mathscr{A}$.

(3″) *Type II adversary*: $\mathscr{B}$ can set revocation list RL^*, $\mathbf{D} = \mathbf{B}_{\mathrm{out}}$, and let $\mathrm{mpk} = ((\mathbf{A}, \mathbf{B}), (\{\mathbf{B}_{i,b}\}_{i\in[l],b\in\{0,1\}}), \mathbf{C}, \mathbf{D}, \mathbf{G})$, and send mpk to the adversary $\mathscr{A}$.

The $\mathscr{B}$ answers $\mathscr{A}$'s query to the $\mathcal{O}$ as follows:

$\mathrm{GenSK}(\cdot)$:

Type I adversary: when queried id^* from $\mathscr{A}$, $\mathscr{B}$ can return $\mathrm{sk}_{\mathrm{id}^*} = R_{\mathrm{id}}^*$. When queried $\mathrm{id} \ne \mathrm{id}^*$ from $\mathscr{A}$, $\mathscr{B}$ can set $\mathbf{F}_{\mathrm{id}} = \mathbf{A}_1 + H(\mathrm{id})\mathbf{G} = \mathbf{A}\mathbf{R}_1 + (H(\mathrm{id}) - (\mathrm{id}^*))\mathbf{G}$, and then run sample algorithm $\mathbf{R}_{\mathrm{id}} \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{R}_1, (H(\mathrm{id}) - H(\mathrm{id}^*))\mathbf{G}, T_{\mathbf{G}}, \mathbf{D}, s)$. Finally, $\mathscr{B}$ can return $\mathrm{sk}_{\mathrm{id}} = \mathbf{R}_{\mathrm{id}}$.

Type II adversary: when queried $\mathrm{id} \neq \mathrm{id}^*$ from $\mathscr{A}$, $\mathscr{B}$ can set $\mathbf{F}_{\mathrm{id}} = \mathbf{A}_1 + H(\mathrm{id})\mathbf{G} = \mathbf{A}\mathbf{R}_1 + (H(\mathrm{id}) - H(\mathrm{id}^*))\mathbf{G}$, and then sample $\mathbf{R}_{\mathrm{id}} \longleftarrow \mathrm{SampleRight}(\mathbf{A}, \mathbf{R}_1, (H(\mathrm{id}) - H(\mathrm{id}^*))\mathbf{G}, T_{\mathrm{G}}, \mathbf{D}, s)$. Finally, $\mathscr{B}$ can return $\mathrm{sk}_{\mathrm{id}} = \mathbf{R}_{\mathrm{id}}$.

GenPK$(\cdot)$: when $\mathscr{A}$ queries GenPK for id and $\mathscr{C}_{\mathrm{id}}$, $\mathscr{B}$ can set $\mathbf{F}_{\mathrm{id}} = \mathbf{A}_1 + H(\mathrm{id})\mathbf{G} = \mathbf{A}\mathbf{R}_1 + (H(\mathrm{id}) - (\mathrm{id}^*))\mathbf{G}$ and $\mathbf{B}_{\mathrm{id}} = \mathbf{B}_1 + H(\mathrm{id})G = \mathbf{B}\mathbf{R}_2 + (H(\mathrm{id}) - (\mathrm{id}^*))\mathbf{G}$. And then $\mathscr{B}$ does the following:

(1) When $\mathscr{A}$ queries GenPK for id^* such that $\mathscr{C}_{\mathrm{id}^*}(\mathrm{att}^*) = 1$, store id^* in leaf node θ from BT and set $\mathbf{F}_{\mathrm{id}}$ as above. If $x \in \mathrm{Path}(\mathrm{id}^*)$, pick $\mathbf{Z}_{1,x} \longleftarrow \mathscr{D}_{\mathbb{Z}^{2m\times m},s}$ and set $\mathbf{U}_x = \mathbf{F}_{\mathrm{id}^*} - [\mathbf{B} \mid \mathbf{B}_{\mathrm{id}^*}]\mathbf{Z}_{1,x}$. And then for the gate $x_{\ell+i} = \mathscr{C}_{\mathrm{id}^*}(x_{u_i}, x_{v_i}, b', b'') \in \{0,1\}^2$, $i = 1, \ldots, |\mathscr{C}_{\mathrm{id}^*}| - \ell$, $(\mathbf{R}^{(u_i,v_i,\ell+i)}_{(b',b'',\mathscr{C}_{\mathrm{id}^*i}(b',b''))}, \mathbf{B}_{\ell+i,C_{\mathrm{id}^*_i}(b',b'')})$. And $\mathscr{B}$ can output $s_{\mathrm{id}^*} = \left\{\mathbf{R}^{(u_i,v_i,\ell+i)}_{(b',b'',\mathscr{C}_{\mathrm{id}^*_i}(b',b''))}, (b',b'') \in \{0,1\}^2, i = 1, \ldots, |\mathscr{C}_{\mathrm{id}^*}| - \ell\right\}$. When $\mathscr{A}$ queries GenPK for id^* and $\mathscr{C}_{\mathrm{id}^*}$, $\mathscr{B}$ can return $\mathrm{pk}_{\mathrm{id}^*} = \left\{s_{\mathrm{id}^*}, \left\{(x, \mathbf{Z}_{1,x})\right\}_{x\in\mathrm{Path}(\mathrm{id}^*)}\right\}$. If $x \notin \mathrm{Path}(\mathrm{id}^*)$, $\mathbf{Z}_{2,x} \longleftarrow \mathscr{D}_{\mathbb{Z}^{2m\times m},s}$, and set $\mathbf{U}_x = [\mathbf{B}\mathbf{C}_{\mathbf{t}^*}]\mathbf{Z}_{2,x}$.

(2) When $\mathscr{A}$ queries GenPK for id such that $\mathscr{C}_{\mathrm{id}}(\mathrm{att}^*) \neq 1$, for $x \in \mathrm{Path}(\mathrm{id})$, $\mathscr{B}$ Sample $\mathbf{Z}_{1,x} \longleftarrow \mathrm{SampleRight}(\mathbf{B}, \mathbf{R}_2, (H(\mathrm{id}) - H(\mathrm{id}^*))\mathbf{G}, T_{\mathbf{G}}, \mathbf{D}, s)$. Note that $[\mathbf{B} \mid \mathbf{B}_{\mathrm{id}}]Z_{1,x} = \mathbf{F}_{\mathrm{id}} - \mathbf{U}_x$. $\mathscr{B}$ can ask $\mathscr{A}$ for a matrix $\mathbf{B}_{\mathrm{id}}$ to run KeyGen by using $\mathscr{C}_{\mathrm{id}}$ and get $s_{\mathrm{id}} = \left\{\mathbf{R}^{(u_i,v_i,\ell+i)}_{\mathbf{B}(b',b'',\mathscr{C}_{\mathrm{id}_i}(b',b''))}, (b',b'') \in \{0,1\}^2, i = 1, \ldots, |\mathscr{C}_{\mathrm{id}}| - \ell\right\}$ such that $\mathscr{B}$ can only get a code of $\mathbf{B}_{\mathrm{id}}$ from s_{id} by using $\left\{\mathbf{B}_{i,b}\right\}_{i\in[l],b\in\{0,1\}}$. That is, $\mathscr{A}$ sets $\mathrm{pk}_{\mathrm{tgt}} = \mathbf{B}_{\mathrm{id}}$. Other than that, $\mathscr{B}$ did not get any secret information. This will not endanger the security of GVW13. Then $\mathscr{B}$ outputs $\mathrm{pk}_{\mathrm{id}} = \left\{s_{\mathrm{id}}, \left\{x, \mathbf{Z}_{1,x}\right\}_{x\in\mathrm{path}(\mathrm{id})}\right\}$.

KeyUp$(\cdot)$: for key update of time $\mathbf{t} \neq \mathbf{t}^*$ and all $x \in \mathrm{KUNodes}(\mathrm{BT}, \mathrm{RL}, \mathbf{t})$, set $\mathbf{C}_t = \mathbf{B}\mathbf{R} + (H(\mathbf{t}) - H(\mathbf{t}^*))\mathbf{G}$. $\mathscr{B}$ can compute ku_t as $\mathbf{Z}_{2,x} \longleftarrow \mathrm{SampleRight}(\mathbf{B}, \mathbf{R}_3, (H(t) - H(t^*))\mathbf{G}, T_{\mathrm{G}}, \mathbf{D}, s)$ where $\mathbf{U}_x$ has been defined in GenPK$(\cdot)$ and return $\mathrm{ku}_t = \left\{(x, \mathbf{Z}_{2,x})\right\}_{x\in\mathrm{KUNodes}(\mathrm{BT},\mathrm{RL}^*,t)}$.

TranKG$(\cdot)$ and Transform$(\cdot)$: by using a key update ku_t and a public key $\mathrm{pk}_{\mathrm{id}}$ with identity id, $\mathscr{B}$ can execute these two algorithms.

Revoke$(\cdot)$: after accepting the query about updating the revocation list on an identity id, a revocation list RL and a state st, the $\mathscr{B}$ adds id to RL, outputs a new RL, and gives it to $\mathscr{A}$.

Then $\mathscr{A}$ gives two message $\mu_0, \mu_1 \in \mathscr{M}$ to $\mathscr{B}$ who prepares the challenge ciphertext as follows:

(1) Send μ_0, μ_1 which are seen as two challenge messages. The $\mathscr{A}$ chooses $\beta \longleftarrow \{0,1\}$ and returns a ciphertext $\mathrm{ct}_{\mathrm{att}^{*\prime}} = (\mathrm{att}^{*\prime}, \gamma, \left\{\varphi_j\right\}_{j\in[\ell+2]})$ as a GVW13's encryption of μ_b under attribute att^*.

(2) Output $\mathrm{ct}_{t^*,\mathrm{att}^*} = (\mathrm{att}^*, \gamma', \varphi', \psi', \xi, \{\psi_i\}_{i\in[\ell]})$ as an SR-ABE ciphertext of μ_β under att^*, t^* where

$$\begin{cases} \gamma' = \gamma, \\ \psi_i = \varphi_i, \quad i \in [\ell], \\ \psi' = \varphi_{\ell+2,0}, \\ \xi = \mathbf{R}_1^T\psi', \\ \varphi' = \varphi_{\ell+1,0}. \end{cases} \tag{11}$$

After being allowed to make additional queries, $\mathscr{A}$ outputs $\beta' \in \{0,1\}$. Then the adversary $\mathscr{B}$ returns it to $\mathscr{A}$ as the guess of the bit $\mathscr{B}$.

Because of assuming that $\mathscr{A}$ can break the selective security of SR – ABE with probability ε, which means

$$\mathrm{Adv}^{\mathrm{SR-ABE}}_{\mathscr{A}}(\lambda, \ell, d_{\max}) = \left|\Pr[\beta' = \beta] - \frac{1}{2}\right| = \varepsilon, \tag{12}$$

then, we have

$$\mathrm{Adv}^{\mathrm{GVW13}}_{\mathscr{B}}(\lambda, \ell, d_{\max}) = \left|\Pr[\beta' = \beta] - \frac{1}{2}\right| = \varepsilon. \tag{13}$$

□

4.3. Comparison. In the past few years, a large body of work on revocable ABE [34, 35] and revocable IBE [37, 39] has been proposed. In these revocable ABE schemes [34, 35], there is a powerful but untrustworthy server. And most of data users' workloads are delegated to the powerful untrusted server such that the KGC indirectly revokes users in revocation list by stopping updating the keys without any operation by the user. In [34], a revocable CP-ABE is proposed where a user can generate its local secret key and public key and decrypt a ciphertext by using the local secret key. And in [35], a key-randomization was introduced such that a user's local decryption keys can be exposed if the user is not revoked. In these revocable IBE schemes [37, 39], the KGC can revoke the users in the revocation list by stopping posting key update for these users, thereby forcing revoked users to be unable to generate their decryption keys. In [37], a revocable IBE from LWE is proposed where users can transform a long-term secret key and a key update from KGC into decryption keys. And in [39], a generic construction of an RIBE scheme with DKER was proposed which consists of any two-level standard HIBE scheme and RIBE scheme without DKER.

Table 1 compares our SR-ABE scheme with revocable ABE/IBE schemes [34, 35, 37, 39]. In Table 1, N denotes the number of all users in system, R denotes the number of users in revocation list, "-" denotes not-applicable or not-comparable. T_{m} denotes the time taken for matrix multiplication, T_g denotes the time running the Gaussian sample, T_{k} denotes the time running Keygen$(\cdot)$, and T_{s} denotes the time

Table 1: Comparisons of our SR-ABE with other revocable schemes.

	CDLQ [34]	QZZC [35]	CLL+ [37]	KMT [39]	Ours
Problem	DBDH	DBDH	LWE	LWE	LWE
Model	CP-ABE	CP-ABE	IBE	IBE	KP-ABE
PQC	No	No	Yes	Yes	Yes
Server	Yes	Yes	—	—	Yes
DKER	No	Yes	No	Yes	No
Encryption time	—	—	$4(T_m + T_g)$	$7(T_m + T_g)$	$(\ell + 4) \cdot (T_m + T_g)$
User's decryption time	—	—	$4T_m$	$6T_m$	$2T_m$
GenSK + GenPK + KeyUp Time	—	—	$T_k + (\log N + R\log(N/R))$	$3T_k + (\log N + R\log(N/R))$	$2\lvert\mathscr{C}_{id}\rvert T_k + (\log N + 1R\log(N/R))$
Server-key size	$O(R\log(N/R))$	$O(R\log(N/R))$	—	—	$O(R\log(N/R))$
User-key size	$O(1)$	$O(1)$	$O(\log N) + O(R\log(N/R))$	$O(\log N) + O(R\log(N/R))$	$O(1)$

running SampleLeft$(\cdot)$. The schemes [34, 35] are based on decisional Bilinear Diffie–Hellman (DBDH) assumption from discrete logarithm problem and insecure when faced with the adversaries using quantum computers. Compared with them, our scheme is based on LWE and secure against the quantum computers. Compared with the schemes [37, 39], in our scheme, KGC needs more computation cost due to the complexity of current strategy function in ABE, but users need less computation cost in decryption. In the schemes [34, 35], storage overhead is $O(\log N)+O(R\log(N/R))$, which is related to the number of users in system and users in revocation list. Our scheme mitigates user's storage overheads by delegating the most of users' workload to a powerful untrusted server. Our goal in this paper is to achieve user revocation in a KP-ABE system from LWE such that most of the user's workload is delegated to a powerful untrusted server and our scheme can be secure against quantum computers.

5. Conclusion

In this paper, we propose a new model called server-aided revocable attribute based encryption (SR-ABE) from lattice to achieve efficient user revocation and security against quantum computers in attribute-based encryption (ABE). We formally define an SR-ABE model and give the definitions of the correctness and security of SR-ABE from LWE. Based on a standard (nonrevocable) ABE [16], we propose the first concrete construction of SR-ABE from lattices. And, we provide a more rigorous proof of security, based on the hardness of LWE.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China under grants no. 2017YFB0802000, National Natural Science Foundations of China (Nos. 61672412 and 61972457), National Cryptography Development Fund under grant no. MMJJ20170104, National Natural Science Foundation of China under Grant nos. U19B2021 and U1736111, National Cryptography Development Fund under Grant no. MMJJ20180111, and Key Foundation of Science and Technology Development of Henan Province (no.202102210356).

References

[1] S. Amit and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, Aarhus, Denmark, 2005.

[2] 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*, pp. 89–98, ACM, Chicago, IL, USA, 2006.

[3] Adi Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pp. 47–53, Springer, Paris, France, April 1984.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[5] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K.-K. R. Choo, "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 72–83, 2019.

[6] L. Allison, T. Okamoto, S. Amit, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 62–91, Springer, French Riviera, Monaco, 2010.

[7] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Proceedings of the Annual Cryptology Conference*, pp. 191–208, Springer, Barbara, CA, USA, August 2010.

[8] X. Boyen, "Attribute-based functional encryption on lattices," in *Theory of Cryptography*, pp. 122–142, Springer, Berlin, Germany, 2013.

[9] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 162–179, Springer, Beijing, China, April 2013.

[10] L. Allison and B. Waters, "New proof methods for attribute-based encryption: achieving full security through selective techniques," in *Annual Cryptology*, pp. 180–198, Springer, Berlin, Germany, 2012.

[11] B. Waters, "Functional encryption for regular languages," in *Annual Cryptology*, pp. 218–235, Springer, Berlin, Germany, 2012.

[12] Z. Brakerski, D. Cash, R. Tsabary, and H. Wee, "Targeted homomorphic attribute-based encryption," in *Theory of Cryptography*, pp. 330–360, Springer, Berlin, Germany, 2016.

[13] D. Boneh, G. Craig, S. Gorbunov et al., "Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 533–556, Springer, Copenhagen, Denmark, May 2014.

[14] Z. Brakerski and V. Vaikuntanathan, "Circuit-abe from lwe: unbounded attributes and semi-adaptive security," in *Proceedings of the Annual International Cryptology Conference*, pp. 363–384, Springer, Santa Barbara, CA, USA, August 2016.

[15] S. Garg, G. Craig, S. Halevi, S. Amit, and B. Waters, "Attribute-based encryption for circuits from multilinear maps," in *Proceedings of the Annual Cryptology Conference*, pp. 479–499, Springer, Santa Barbara, CA, USA, August 2013.

[16] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Attribute-based encryption for circuits," in *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, pp. 545–554, ACM, Palo Alto, CA, USA, June 2013.

[17] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th*

ACM Conference on Computer and Communications Security, pp. 417–426, ACM, Alexandria, VA, USA, October 2008.

[18] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the Annual International Cryptology Conference*, pp. 41–62, Springer, Santa Barbara, CA, USA, August 2001.

[19] B. Libert and D. Vergnaud, "Adaptive-ID secure revocable identity-based encryption," in *Proceedings of the Cryptographers Track at the RSA Conference*, pp. 1–15, Springer, San Francisco, CA, USA, April 2009.

[20] 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, Nara, Japan, February 2013.

[21] J. H. Seo and K. Emura, "Revocable identity-based cryptosystem revisited: security models and constructions," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1193–1205, 2014.

[22] Against Insiders, "Revocable hierarchical identity-based encryption: history-free update, security against insiders, and short ciphertexts," in *Proceedings of the Topics in Cryptology—CT-RSA 2015: The Cryptographer's Track at the RSA Conference*, vol. 9048, p. 106, Springer, San Francisco, CA, USA, April 2015.

[23] J. H. Seo and K. Emura, "Revocable hierarchical identity-based encryption via history-free approach," *Theoretical Computer Science*, vol. 615, pp. 45–60, 2016.

[24] X. Mao, J. Lai, K. Chen, J. Weng, and Q. Mei, "Efficient revocable identity-based encryption from multilinear maps," *Security and Communication Networks*, vol. 8, no. 18, pp. 3511–3522, 2015.

[25] S. Park, K. Lee, and D. H. Lee, "New constructions of revocable identity-based encryption from multilinear maps," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1564–1577, 2015.

[26] Y. Ishida, J. Shikata, and Y. Watanabe, "CCA-secure revocable identity-based encryption schemes with decryption key exposure resistance," *International Journal of Applied Cryptography*, vol. 3, no. 3, pp. 288–311, 2017.

[27] K. Lee, D. H. Lee, and J. H. Park, "Efficient revocable identity-based encryption via subset difference methods," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 39–76, 2017.

[28] Y. Park, K. Emura, and J. H. Seo, "New revocable ibe in prime-order groups: adaptively secure, decryption key exposure resistant, and with short public parameters," in *Proceedings of the Cryptographers Track at the RSA Conference*, pp. 432–449, Springer, San Francisco, CA, USA, March 2017.

[29] B. Qin, R. H. Deng, Y. Li, and S. Liu, "Server-aided revocable identity-based encryption," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 286–304, Springer, Vienna, Austria, September 2015.

[30] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Proceedings of the IMA International Conference on Cryptography and Coding*, pp. 278–300, Springer, Cirencester, UK, December 2009.

[31] 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*, pp. 261–270, ACM, Beijing, China, April 2010.

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

[33] 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 16th International Conference on Information Security*, vol. 7807, pp. 293–308, Springer-Verlag New York, Inc., Dallas, TX, USA, 2013.

[34] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attribute-based encryption," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 570–587, Springer, Heraklion, Greece, September 2016.

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

[36] 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 International Conference on Applied Cryptography and Network Security*, pp. 516–534, Springer, London, UK, 2018.

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

[38] A. Takayasu and Y. Watanabe, "Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance," in *Proceedings of the Australasian Conference on Information Security and Privacy*, pp. 184–204, Springer, Auckland, New Zealand, July 2017.

[39] S. Katsumata, T. Matsuda, and A. Takayasu, "Lattice-based revocable (hierarchical) ibe with decryption key exposure resistance," in *Proceedings of the IACR International Workshop on Public Key Cryptography*, pp. 441–471, Springer, Beijing, China, April 2019.

[40] S. Ling, K. Nguyen, H. Wang, and J. Zhang, "Server-aided revocable predicate encryption: formalization and lattice-based instantiation," 2018, http://arxiv.org/abs/1801.07844.

[41] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan, "Functional encryption for inner product predicates from learning with errors," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 21–40, Springer, Seoul, South Korea, December 2011.

[42] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 99–108, ACM, Philadephia, PA, USA, 1996.

[43] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the ACM Symposium on Theory of Computing*, Baltimore, MD, USA, 2005.

[44] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (h) ibe in the standard model," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 553–572, Springer, Tallinn, Estonia, May 2010.

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

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

[47] M. Ajtai, "Generating hard instances of the short basis problem," in *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pp. 1–9, Springer, Prague, Czech Republic, July 1999.

[48] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe," in *Proceedings of the Annual Cryptology Conference*, pp. 98–115, Springer, Barbara, CA, USA, August 2010.

[49] 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.