Hindawi
Security and Communication Networks
Volume 2020, Article ID 2345369, 11 pages
https://doi.org/10.1155/2020/2345369

*Research Article*

# Space-Efficient Key-Policy Attribute-Based Encryption from Lattices and Two-Dimensional Attributes

**Yuan Liu,[1] Licheng Wang,[1] Xiaoying Shen,[1] Lixiang Li,[1] and Dezhi An[2]**

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*School of Cyber Security, Gansu University of Political Science and Law, Lanzhou 730070, China*

Correspondence should be addressed to Licheng Wang; wanglc2012@126.com

Received 24 February 2020; Revised 13 May 2020; Accepted 16 July 2020; Published 7 August 2020

Academic Editor: Fulvio Valenza



Linear secret-sharing scheme (LSSS) is a useful tool for supporting flexible access policy in building attribute-based encryption (ABE) schemes. But in lattice-based ABE constructions, there is a subtle security problem in the sense that careless usage of LSSS-based secret sharing over vectors would lead to the leakage of the master secret key. In this paper, we propose a new method that employs LSSS to build lattice-based key-policy attribute-based encryption (KP-ABE) that resolves this security issue. More specifically, no adversary can reconstruct the master secret key since we introduce a new trapdoor generation algorithm to generate a strong trapdoor (instead of a lattice basis), that is, the master secret key, and remove the dependency of the master secret key on the total number of system attributes. Meanwhile, with the purpose of reducing the storage cost and support dynamic updating on attributes, we extended the traditional 1-dimensional attribute structure to 2-dimensional one. This makes our construction remarkably efficient in space cost, with acceptable time cost. Finally, our scheme is proved to be secure in the standard model.

## 1. Introduction

In 2005, Sahai and Waters [1] proposed a new public key encryption mechanism: attribute-based encryption (ABE). It associates the user's identity with a set of attributes. The user's private key and the ciphertext are defined based on the attribute set and access policy, respectively, and a user can decrypt only if the attribute set and the access policy match each other. If the user's privacy key is correlated to the access policy and the ciphertext is correlated to the attribute set, it is a key-policy ABE (KP-ABE). On the contrary, it is a ciphertext-policy ABE (CP-ABE) [2]. With the development of the cloud computing, more and more people tend to store and share their data through the cloud. To protect the users' privacy information, ABE is a good choice which can achieve fine-grained access control and one-to-many communication of users' data in public cloud storage [3].

The constructions of ABE are usually based on two different mathematical platforms: bilinear pairings and lattices. On the one hand, the quick progress in pairing-based ABE constructions [4–7] fosters the so-called expressive cryptography. On the other hand, with the breakthrough of quantum computing technology in recent years, most researchers believe that bilinear pairing-based constructions suffer from the potential threat of quantum computers. Therefore, the study of lattice-based ABE schemes attracts more attention.

In 2011, Zhang and Zhang [8] proposed a CP-ABE scheme that is based on learning with errors (LWE) problem [9] and supports AND operation among attributes. One year after, they [10] again proposed another lattice-based CP-ABE scheme that supports threshold access policies. In 2013, Liu et al. [11] proposed a threshold ABE scheme with attribute hierarchy based on lattice intractability assumptions. The constructions in [8, 10, 11] gave us a good inspiration to study lattice-based ABE. But a single AND operation or THRESHOLD operation is still not enough for describing even flexible access policies in practical application. To

support an even flexible access policy, the technique of linear secret-sharing scheme (LSSS) which is known to support AND, OR, and THRESHOLD operations is useful. For instance, LSSS was used in building pairing-based ABE schemes [12–14]. However, as mentioned by Agrawal et al. [15], in building a lattice-based KP-ABE scheme, if the usage of LSSS is improper (such as work in [15]), the shares of a vector are correlated, and this would enable an adversary to reconstruct the master secret key by making some correlated key queries. For example, the adversary can make key queries for $(a_1 \wedge a_2) \vee a_3$, $(a_1 \wedge a_2) \vee a_4$, $(a_1 \wedge a_2) \vee a_5$, and so on. Their preimages (i.e., secret key) can be combined to form a short vector in the null space. And after several key queries, the adversary can construct a full basis that can be used to break the challenge ciphertext for a target attribute vector such as $110 \ldots 00$. To deal with this subtle problem, Boyen [16] in their lattice-based KP-ABE construction utilized the LSSS technique but bypassed the vector-based secret sharing. Instead, they constructed a virtual encryption matrix which cascaded the sharing matrix. In 2014, Boneh et al. [17] proposed another lattice-based KP-ABE scheme that uses the arithmetic circuits to describe the access policy which can support AND and OR operations on the attributes.

Another issue of designing ABE schemes is the dimension of structure of attributes. The typical setting is to use 1-dimensional structure. That is, a set $\{1, \ldots, \ell\}$ is used to denote the system attributes and all attributes of the system are initialized in Setup phase (such as in [16–18]). However, in [16–18], this kind of setting has two problems in practice. First, the number of system attributes is fixed. Or equivalently, the attributes space is bounded after the execution of the Setup algorithm. Second, the size of parameters would be linearly increased with the number of system attributes. This situation becomes ever worse in lattice-based ABE constructions. The total size of parameters would be over 300 M bytes, even 1 G byte. In particular, the space cost for attribute parameters occupied over 95% of the space cost of the total public parameters. But in practical application, it is more desirable to support dynamic updating and space-efficient settings on system attributes. Therefore, the main motivation of this paper is to construct a lattice-based KP-ABE scheme that supports flexible access policies and dynamic updating and space-efficient attribute settings.

*1.1. Our Contribution.* In this paper, we propose a secure LSSS-based KP-ABE scheme from lattices which can support a flexible access policy but has solved the master secret key leakage problem. In addition, we give a flexible attribute description which can add new attributes dynamically and we also reduce the sizes of public parameter, master key, user's secret key, and ciphertext. The main contributions are as follows:

(1) New method of LSSS-enabled flexible access policy without the security issue mentioned by Agrawal. In our construction, we use the LSSS technique to support a flexible access policy and resolve the insecure problem by avoiding the adversary to reconstruct the master secret key. In the previous LSSS-based KP-ABE scheme, the master secret key often consisted of some bases which correlated to the total number of system attributes. An adversary can reconstruct a full basis (i.e., master secret key) by making some correlated key queries [15]. But in our scheme, we use a new trapdoor generation algorithm of MP12 [19] to generate a strong trapdoor, not a basis (i.e., the master secret key), and the master secret key is no longer correlated to the number of system attributes. Thus, no adversary can obtain the correlated information of the master key. Since the master key is a new strong trapdoor (instead of a lattice basis), we also make an improvement of the SampleLeft algorithm which takes a lattice basis as input (see Section 4.1).

(2) Two-dimensional attribute structures that support attribute dynamic updating and reduce size of parameters. In our scheme, we extended the traditional 1-dimensional *(value-specified)* attribute structures to 2-dimensional *(label- and value-specified)* ones. Multiple values can be set under an attribute label. It can add new attribute values at any time without reconstructing the system. The attribute space is no longer bounded. In addition, the attribute value usually often contains more privacy information than attribute label. By doing that, the attribute labels are used to set the access policy, while the actual values are hidden. So even if the adversary makes some key queries for correlated access policy, it cannot get any privacy information of the master secret key. Particularly, the most observable advantage of using this kind of 2-dimensional attribute structure is that the storage cost is reduced. The storage cost of the trapdoor, not a basis, is at least four times smaller than other constructions. And by removing the reliance on the total number of system attributes, the sizes of public parameter, master secret key, user's secret key, and ciphertext are remarkably reduced. The detailed performance analysis is given in Section 5. In fact, this kind of 2-dimensional settings on attributes is now new to us. In 2015, Ying et al. [13] used a tag-based setting on attributes in pairing-based constructions. Our contribution is to introduce Ying et al.'s idea into lattice-based constructions.

*1.2. Related Work.* In 2012, Agrawal et al. proposed a fuzzy identity-based encryption scheme from lattices [15] which can support a single THRESHOLD operation. In 2014, a lattice-based KP-ABE scheme [20] and a lattice-based CP-ABE scheme [21] were proposed. These two schemes just only supported AND gate. In 2015, Zhang et al. [22] designed a multiauthority attribute-based encryption (MA-ABE) scheme on lattice which can support "THRESHOLD" operation on the attributes. There exist multiple attribute

authorities in this scheme which has resolved the problem of delayed response in the single attribute authority. In 2017, Zhao and Gao proposed an LSSS matrix-based KP-ABE on lattices [23] with a flexible access policy, but the attribute space is bounded. Liu et al. proposed a multiauthority key-policy attribute-based encryption (MA-KP-ABE) scheme from lattices [24] with the same advantage of Zhang's construction in [22]. In 2018, Liu et al. [25] proposed an MA-ABE with ciphertext policy. In the scheme, the size of ciphertext is reduced because they introduced the basis delegation without dimension increase algorithm in [26]. The lattice-based ABE schemes in [27, 28] were based on the ring-LWE problem. The authors of [27, 28] make an engineering implement of their schemes.

*1.3. Organization.* The rest of this paper is organized as follows. In Section 2, we give the basic definition of lattice, sampling algorithms, and FRD function. The definition of the KP-ABE scheme and security model are given in Section 3. In Section 4, an improved SampleLeft algorithm and a space-efficient KP-ABE scheme with 2-dimensional attributes are proposed. The parameters setting and security proof are also given in Section 4. In Section 5, we give a detailed comparison between our scheme and relevant works from the space cost and time cost. Finally, we conclude this paper in Section 6.

## 2. Preliminaries

*Notation.* $\mathbb{Z}_q$ denotes an integer set of mod $q$ residue class. $u \in \mathbb{Z}_q^n$ is a $n$-dimension column vector. An $n \times m$ matrix is denoted by $A \in \mathbb{Z}_q^{n\times m}$, where $A = (a_1, \ldots, a_m)$. $\|A\|$ denotes the $\ell_2$-norm length of the longest column of $A$. $\widetilde{A}$ denotes the Gram-Schmidt orthogonalization of the vectors $a_1, \ldots, a_m$. We refer to $\|\widetilde{A}\|$ as the Gram-Schmidt norm of $A$. In our scheme, $A_u$ and $A_c$ denote user's attribute set and ciphertext attributes set, respectively, and $|A_u|$ and $|A_c|$ denote the number of attributes in $A_u$ and $A_c$, respectively.

### 2.1. Integer Lattice

*Definition 1.* $b_1, b_2, \ldots, b_n \in \mathbb{R}^m$ are $n$ linearly independent vectors, and the lattice $\Lambda$ is generated by the following formula:

$$\Lambda = L(B) = \left\{ \sum_{i=1}^{n} x_i b_i \colon x_i \in \mathbb{Z}, (i = 1, \ldots, n) \right\}. \quad (1)$$

Note that $B = [b_1, b_2, \ldots, b_n]$ is a basis of $\Lambda$, $n$ is the rank, and $m$ is the dimension.

*Definition 2.* For prime $q$, $A \in \mathbb{Z}_q^{n\times m}$, and $u \in \mathbb{Z}_q^n$, define

$$\begin{aligned} \Lambda_q(A) &= \left\{ y \in \mathbb{Z}^m \text{s.t. } \exists s \in \mathbb{Z}_q^n A^\top s = y \,(\mathrm{mod}\, q) \right\}, \\ \Lambda_q^\perp(A) &= \left\{ y \in \mathbb{Z}^m \text{s.t. } Ay = 0 \,(\mathrm{mod}\, q) \right\}, \\ \Lambda_q^u(A) &= \left\{ y \in \mathbb{Z}^m \text{s.t. } Ay = u \,(\mathrm{mod}\, q) \right\}. \end{aligned} \quad (2)$$

### 2.2. Discrete Gaussians

*Definition 3.* For a positive integer $s \in \mathbb{R}$ and a vector $c \in \mathbb{R}^m$, we defined a Gaussian distribution with center $c$ and variance $s$ as follows:

$$\mathscr{D}_{\Lambda,\sigma,c} = \frac{\rho_{\sigma,c}(x)}{\rho_{\sigma,c}(\Lambda)} = \frac{\rho_{\sigma,c}(x)}{\sum_{x\in\Lambda}\rho_{\sigma,c}(x)}, \quad (3)$$

where $\sigma > 0$ is a parameter and $\rho_{\sigma,c}(x) = \exp(-\pi(\|x - c\|^2/\sigma^2))$.

*2.3. Sampling Algorithms and FRD Function.* The following two algorithms are introduced from MP12 [19]. *TrapGen* is used to generate the public parameters and master key in the KP-ABE scheme.

*Definition 4.* A $G$-trapdoor for $A$ is a matrix $T_A \in \mathbb{Z}^{(m-\overline{n})\times\overline{n}}$ such that $A\begin{bmatrix} T_A \\ I \end{bmatrix} = HG$, where $A \in \mathbb{Z}_q^{n\times m}$, $G \in \mathbb{Z}_q^{n\times\overline{n}}$, and $H \in \mathbb{Z}_q^{n\times n}$. $H$ is a tag of the trapdoor. $s_1(T_A)$, the largest singular value of $T_A$, is the quality of the trapdoor.

For any integer $q \geq 2, n \geq 1, \overline{n} = nt, t = \lceil \log q \rceil$ and sufficiently large $m = O(n \log q)$, let $H \in \mathbb{Z}_q^{n\times n}$ be a invertible matrix, $G \in \mathbb{Z}_q^{n\times\overline{n}}$ denotes a primitive and public matrix, $\overline{A} \in \mathbb{Z}_q^{n\times\overline{m}}$ is chosen at random, $u \in \mathbb{Z}_q^n$ is a uniformly random vector, and $\sigma \geq s_1(T_A)\|\widetilde{G}\|$ is the Gaussian parameter, where $s_1(T_A)$ is the largest singular value of $T_A$ and $\|\widetilde{G}\|$ is the maximum length of $G$'s Gram-Schmidt orthogonalized vectors ($\|\widetilde{G}\| = 2$ or $\sqrt{5}$ [19]); it has the following:

(1) Algorithm $TrapGen(A, H)$ that outputs a uniformly random matrix $A = [\overline{A}|HG - \overline{A}T_A] \in \mathbb{Z}_q^{n\times m}$ and a trapdoor matrix $T_A \in \mathbb{Z}^{\overline{m}\times\overline{n}}$, where the trapdoor size is $s_1(T_A) = s \cdot O(\sqrt{\overline{m}} + \sqrt{\overline{n}})$ $(s > 0)$ and $m = \overline{m} + \overline{n}$.

(2) Algorithm $SamplePre(A, G, T_A, u, \sigma)$ that outputs a vector $e \in \mathbb{Z}_q^{\overline{m}+\overline{n}}$, where $Ae = u \,(\mathrm{mod} q)$.

**Lemma 1.** *The vector e in (2) is not statistically distinguishable from* $\mathscr{D}_{\Lambda_q^u(A),\sigma\omega(\sqrt{\log n})}$, *where [19]*

$$\Pr\left[ e \sim \mathscr{D}_{\Lambda_q^u(A),\sigma\omega(\sqrt{\log n})} \colon \|e\| > \sigma\sqrt{m} \right] \leq negl(n). \quad (4)$$

The following *SampleRight* algorithm is used in our scheme for the security proof. There also exists a *SampleLeft* algorithm. But in the traditional *SampleLeft* algorithm in [29], $T_A$ is a basis of $\Lambda_q^\perp(A)$. But since in our scheme the trapdoor $T_A$ is a trapdoor, not a basis, we make a small improvement to this algorithm (see Section 4.1). We call the improved SampleLeft algorithm *IMSampleLeft*.

For $q > 2$, $m > n$, and $\sigma > \|\widetilde{T}_B\| \cdot s_R\omega(\sqrt{\log m})$, algorithm *SampleRight* $(A, B, R, T_B, u, \sigma)$ outputs a vector $e \in \mathbb{Z}^{m+k}$, where $A \in \mathbb{Z}_q^{n\times k}$, $B \in \mathbb{Z}_q^{n\times m}$, $R \in \mathbb{Z}_q^{k\times m}$, and $T_B$ is a basis of $\Lambda_q^\perp(B)$ and $u \in \mathbb{Z}^n$. The vector $e$ is not statistically distinguishable from $\mathscr{D}_{\Lambda_q^u(F_2),\sigma}$, where $F_2 = A|AR + B$ and $F_2 \cdot e = u \,(\mathrm{mod} q)$. Note that the matrix $R$ often is a random matrix in $\{-1, 1\}^{m\times m}$, $\|\widetilde{T}_B\|$ is the maximum length of $T_B$'s Gram-Schmidt orthogonalized vectors, and $s_R = \|\widetilde{R}\| < O(\sqrt{m})$.

The following algorithm is the encoding with full-rank differences (FRD) function. For a prime $q$ and a positive integer $n$, a FRD function $H$: $\mathbb{Z}_q^n \longrightarrow \mathbb{Z}_q^{n\times n}$ is as follows:

(1) FRD: for an input $u = (u_0, u_1, ..., u_{n-1})^\top \in \mathbb{Z}_q^n$, define the polynomial $g_u(x) = \sum_{i=0}^{n-1} u_i x^i$. Define $H(u)$ as

$$H(u) = \begin{bmatrix} coe(g_u) \\ coe(x \cdot g_u \bmod f) \\ \vdots \\ coe(x^{n-1} \cdot g_u \bmod f) \end{bmatrix} \in \mathbb{Z}_q^{n\times n}. \quad (5)$$

(2) Let $f$ be some polynomial of degree $n$ that is irreducible and $coe(g)$ denote the $n$ vector of coefficients of $g$.

*2.4. Two Lemmas to Bound Norms.* The following three lemmas will be used to prove that the decryption is correct.

**Lemma 2.** *Let $e$ be some vector in $\mathbb{Z}^m$ and let $y \xleftarrow{R} \overline{\Psi}_\alpha^m$ [30]. Then the quality $|e^\top y|$ treated as an integer in $[0, q-1]$ satisfies*

$$|e^\top y| \le \|e\| q\alpha\omega(\sqrt{\log m}) + \|e\|\sqrt{m}/2, \quad (6)$$

*with all but negligible probability in $m$.*

As a special case, Lemma 2 shows that if $x \xleftarrow{R} \overline{\Psi}_\alpha$ is treated as an integer in $[0, q-1]$, it satisfies

$$|x| \le q\alpha\omega(\sqrt{\log m}) + \frac{1}{2}, \quad (7)$$

with all but negligible probability in $m$.

**Lemma 3.** *For a vector $u \in \mathbb{R}^m$ and a random matrix $R \in \{-1, 1\}^{k\times m}$ [29], it has*

$$\Pr[\|Ru\| > \sqrt{k}\,\omega(\sqrt{\log k})] < negl(k). \quad (8)$$

*2.5. Linear Secret-Sharing Scheme (LSSS).* A secret-sharing scheme over a collection $P$ is linear if one has the following:

(1) The shares for each party form a vector over $\mathbb{Z}_q$.

(2) There exists a matrix $M$ of size $l \times n$ such that, for all $i = 1, ..., l$, the $i$'th row is labeled with a function $\rho(i)$. Randomly choose $s \in \mathbb{Z}_q$ and a vector $g = (s, g_2, ..., g_n)^\top \in \mathbb{Z}_q^n$, where $s$ is the secret to be shared. The share $\lambda_i = M_i g$ belongs to party $\rho(i)$, where $M_i$ is the i'th row of $M$.

Linear reconstruction property: suppose a scheme's access structure is LSSS. Let $S$ be an authorized set and $I = \{i \mid \rho(i) \in S\}$. There exists a set of constants $\{k_i \in \mathbb{Z}_q\}_{i\in I}$ that can be used to compute the secret $s$: $\sum_{i\in I} k_i \lambda_i = s$.

*2.6. Hardness Assumption*

*Definition 5.* Give a prime $q$, a positive integer $n$, and a distribution $\overline{\Psi}_\alpha$ over $\mathbb{Z}_q$. A $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being either a noisy pseudorandom sampler $\mathcal{O}_s$ carrying some constant random secret key $s \in \mathbb{Z}_q$ or a truly random sampler $\mathcal{O}_s'$, whose behaviors are as follows, respectively:

$\mathcal{O}_s$ outputs samples of the form $(w_i, v_i) = (w_i, w_i^\top s + \chi_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $w_i$ is uniform in $\mathbb{Z}_q^n$, $s \in \mathbb{Z}_q^n$ is a uniformly distributed secret key, and $\chi_i$ is a noise component from $\overline{\Psi}_{\alpha_i}$.

$\mathcal{O}_s'$ outputs truly uniform random samples $(w_i, v_i)$ from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

The $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem allows a number of queries to the challenge oracle $\mathcal{O}$. We say an algorithm $\mathcal{A}$ decides a $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem if $Adv[\mathcal{A}] = |\Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_s'} = 1]|$ is nonnegligible for a random $s \in \mathbb{Z}_q^n$.

**Theorem 1.** *If there exists an efficient, possibly quantum, algorithm for deciding the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem for $q > 2\sqrt{n}/\alpha$, then there is an efficient quantum algorithm for approximating the SIVP and GapSVP problems to within $\tilde{O}(n/\alpha)$ factors in the $\ell_2$ norm, in the worst case [9].*

## 3. Definitions of KP-ABE and Security Model

In KP-ABE, the message is encrypted by using the attributes as public keys, and a user's private key is related to the access policy which is defined by a set of attributes. A KP-ABE scheme consists of the following four algorithms.

*Setup* $(1^n) \longrightarrow (pp, msk)$. Taking a security parameter $1^n$ as input, the algorithm outputs the public parameter $pp$ and the master secret key $msk$.

*KeyGen* $(pp, msk, A_u, (M, \rho)) \longrightarrow sk_u$. On input of the public parameter $pp$, the master key $msk$, a user's attribute set, and access policy $(M, \rho)$ of which the rows are associated with the attribute labels, the algorithm *KeyGen* outputs the secret keys $sk_u$.

*Encrypt* $(pp, A_c, b) \longrightarrow c$. Taking the public parameter $pp$, the encrypted attribute set $A_c \subseteq U$, and a plaintext bit $b \in \{0, 1\}$ as input, this algorithm outputs the ciphertext $c$.

*Decrypt* $(pp, c, sk_u) \longrightarrow b$. On input of the public parameter $pp$, the ciphertext $c$, and a decryptor's secret key $sk_u$, this algorithm outputs a message $b$.

*Correctness.* For a user's attributes set $A_u$, all message $b$, and the ciphertext $c \leftarrow$ Encrypt $(pp, A_c, b)$, we have $\Pr[Decrypt(pp, sk_u, c) = b] = 1 - negl(n)$ if $A_u$ and $A_c$ match each other.

Here, we give the definition of the security model which is adapted from [16]. A KP-ABE scheme is secure under the selective attribute and chooses plaintext attack. It can be described by a game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ as follows:

*Target.* The adversary $\mathcal{A}$ announces to $\mathcal{B}$ the challenge attribute set.

*Setup*. $\mathcal{B}$ runs the *Setup* algorithm and sends the public parameter to $\mathcal{A}$.

*Queries*. In this step, $\mathcal{A}$ makes queries for the privacy keys adaptively for the access policy $(M, \rho)$ that the target attribute set does not satisfy. $\mathcal{B}$ answers the queries.

*Challenge*. $\mathcal{A}$ gives a signal that it is ready to accept the challenge. Then it selects a message $b \in \{0, 1\}$ and sends the message to $\mathcal{B}$. The simulator $\mathcal{B}$ responds with a ciphertext which is encrypted under the target attribute set.

*Continuation*. After having obtained the challenge ciphertext, $\mathcal{A}$ is allowed to make repeat for the privacy key queries.

*Decision*. The adversary $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$ and the advantage of the adversary $\mathcal{A}$ in attacking KP-ABE scheme as $\varepsilon = |\Pr[b' = b] - (1/2)|$.

## 4. Secure and Efficient KP-ABE Scheme Form Lattices

*4.1. Improved SampleLeft Algorithm.* In the traditional SampleLeft algorithm [29], $T_A$ is a basis of $\Lambda_q^{\perp}(A)$. But in the our KP-ABE scheme, $T_A$ is a trapdoor, not a basis. So we make a small improvement to this SampleLeft algorithm. We call the improved SampleLeft algorithm *IMSampleLeft*.

For $q > 2, m > n$, and $\sigma \geq s_1(T_A)\|\widetilde{G}\| \cdot \omega(\sqrt{\log(m + m_1)})$, where $s_1(T_A)$ is the largest singular value of $T_A$ and $\|\widetilde{G}\| = 2$ or $\sqrt{5}$, algorithm *IMSampleLeft* $(A, G, M_1, T_A, u, \sigma)$ outputs a vector $e \in \mathbb{Z}^{m+m_1}$, where $A \in \mathbb{Z}_q^{n\times m}$, $G \in \mathbb{Z}_q^{n\times \overline{n}}$, $M_1 \in \mathbb{Z}_q^{n\times m_1}$, $T_A \in \mathbb{Z}_q^{\overline{m}\times\overline{n}}$ is a trapdoor of $A$, and $u \in \mathbb{Z}^n$. The vector $e$ is not statistically distinguishable from $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$, where $F_1 = A|M_1$ and $F_1 \cdot e = u(\mathrm{mod}q)$.

For completeness, we describe the algorithm in detail.

(1) Sample a random vector $e_2 \in \mathbb{Z}^{m_1}$ distributed statistically close to $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma}$.

(2) Let $y = u - (M_1 \cdot e_2)$ do the following:

Choose a Gaussian perturbation $e_1' \in \mathbb{Z}^m$.

Recall to Definition 4; let $H = I$, then we have $A\begin{bmatrix} T_A \\ I \end{bmatrix} = G$.

Let $y\prime = y - Ae_1'$. Sample a Gaussian $\mathbf{z}$ from $\Lambda_y^{\perp}(G)$ and produce $\begin{bmatrix} T_A \\ I \end{bmatrix} z = e_1''$. Note that $Ae_1'' = A\begin{bmatrix} T_A \\ I \end{bmatrix} z = Gz = y\prime$.

Construct $e_1 = e_1' + e_1'' \in \mathbb{Z}^m$, where $Ae_1 = A(e_1' + e_1'') = Ae_1' + y\prime = y$.

(3) Output $e \leftarrow (e_1, e_2) \in \mathbb{Z}^{m+m_1}$.

*4.2. Space-Efficient KP-ABE from Lattices and Two-Dimensional Attributes.* In our scheme, all the universe attribute can be expressed by $l$ attribute labels $U = \{1, 2, ..., l\}$; each attribute label has different attribute values. In the system, $A_u = \{i: a_i\}$ denotes the user attribute set. $i$ denotes the attribute label and $a_i \in \mathbb{Z}_q^n$ is the attribute value with some privacy information.

*Setup* $(1^n) \longrightarrow (pp, msk)$. Taking a security parameter $1^n$ as input, the algorithm outputs the public parameter $pp$ and the master secret key $msk$.

(1) The system firstly executes the $TrapGen(A, H)$ algorithm in MP12 to generate a uniformly random matrix $A \in \mathbb{Z}_q^{n\times m}$ and a trapdoor matrix $T_A \in \mathbb{Z}^{\overline{m}\times\overline{n}}$, where $m = \overline{m} + \overline{n}$.

(2) For $l$ attribute labels in $U$, it chooses $l$ uniformly random matrices $A_1$, $A_2$, ..., $A_l$.

(3) Then it chooses a uniformly random vector $u = (u_1, u_2, ..., u_n)^{\top} \in \mathbb{Z}_q^n$ and a uniformly random matrix $B \in \mathbb{Z}_q^{n\times m}$.

$pp = \{A, B, (A_i)_{i\in U}, u\}$, $msk = \{(T_A)\}$.

*KeyGen* $(pp, T_A, (M, \rho)) \longrightarrow sk_u$. On input of the public parameter $pp$, the master key $msk$, and the access structure $(M, \rho))$, where $M$ is an $l \times n$ share matrix and $\rho$ is a function which maps each row $M_i$ to the attribute labels based on the attribute value $a_i$ in $A_u$, do the following:

(1) Construct $n$ vectors $g_1 = (u_1, g_{12}, g_{13}, ..., g_{1n})^{\top}$, $g_2 = (u_2, g_{22}, g_{23}, ..., g_{2n})^{\top}$,..., and $g_n = (u_n, g_{n2}, g_{n3}, ..., g_{nn})^{\top}$, where $u_1, u_2, ..., u_n$ are the corresponding components of $u$.

(2) Compute $Mg_1 = (\lambda_1^{(1)}, \lambda_2^{(1)}, ..., \lambda_l^{(1)})^{\top}$,..., and $Mg_n = (\lambda_1^{(n)}, \lambda_2^{(n)}, ..., \lambda_l^{(n)})^{\top}$.

(3) For each attribute value $a_i \in A_u$, let $\lambda_i = (M_i g_1, ..., M_i g_n)^{\top} = (\lambda_i^{(1)}, ..., \lambda_i^{(n)})^{\top}$. Then compute $H(a_{\rho(i)})B$.

(4) Let $F_{\rho(i)} = A|A_{\rho(i)} + H(a_{\rho(i)})B$. Then execute the algorithm *IMSampleLeft* $(A, G, A_{\rho(i)} + H(a_{\rho(i)})B, T_A, \lambda_i, \sigma_i)$ to generate the user's secret key $e_{\rho(i)} \in \mathbb{Z}^{2m}$; note that it has $F_{\rho(i)}e_{\rho(i)} = \lambda_i(\mathrm{mod}q)$.

The secret key is $sk_u = \left\{(e_{\rho(i)})_{i\in A_u}\right\}$.

*Encrypt* $(pp, A_c, b) \longrightarrow c$. Taking the public parameter $pp$, the encrypted attribute set $A_c \subseteq U$, and a plaintext bit $b \in \{0, 1\}$ as input, do the following:

(1) Let $L = (l!)^2$. Choose a uniformly random vector $s \in \mathbb{Z}_q^n$ and noise terms $\chi \xleftarrow{\bar{\Psi}_\alpha} \mathbb{Z}_q$ and $\chi_i \xleftarrow{\bar{\Psi}_{\alpha_i}^m} \mathbb{Z}_q^m$.

(2) Choose a uniformly random matrix $R_i \in \{-1, 1\}^{m\times m}$; let $y_i = R_i\chi_i$.

(3) Compute

$$c_0 = u^{\top}s + L\chi + b\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q, \tag{9}$$

$$c_i = F_i^{\top}s + L\begin{bmatrix} \chi_i \\ y_i \end{bmatrix} \in \mathbb{Z}_q^{2m}. \tag{10}$$

The ciphertext is $c = \left\{c_0, (c_i)_{i\in A_c}\right\}$.

*Decrypt* $(pp, c, sk_u) \longrightarrow b$. On input of the public parameter $pp$, the ciphertext $c$, and a decryptor's secret key $sk_u$, do the following:

(1) Let $I$ denote the subset of matched attribute. The decryptor can find a vector $k = (k_1, k_2, ..., k_l)^\top$ such that $\sum_{i\in(1,...,l)} k_i M_i = (1, 0, ..., 0), \forall i \in U: (i \in I) \vee (k_i = 0)$, that is, $\sum_{i\in(1,...,l)} k_i \lambda_i = u$.

(2) Set the Gaussian parameter $\overline{\sigma}_i = \sigma_i \sqrt{m}\omega(\sqrt{\log m})$. Compute

$$b' = c_0 - \sum_{i\in I} k_i e_i^\top c_i \,(\mathrm{mod} q). \tag{11}$$

(3) If $|b' - \lfloor q/2 \rfloor| < \lfloor q/4 \rfloor$, output 1. Otherwise, output 0.

*4.3. Correctness.* In order to ensure the correctness of decryption, we need to ensure that the error term is less than $q/5$ with overwhelming probability (w.h.p.). As we know,

$$\begin{aligned} b' &= c_0 - \sum_{i\in I} k_i e_i^\top c_i \,(\mathrm{mod} q) \\ &= u^\top s + L\chi + b\lfloor \frac{q}{2} \rfloor - \sum_{i\in I} k_i e_i^\top \left( F_i^\top s + L \begin{bmatrix} \chi_i \\ y_i \end{bmatrix} \right) (\mathrm{mod} q) \\ &= b\lfloor \frac{q}{2} \rfloor + \left( u^\top s - \sum_{i\in I} (k_i F_i e_i)^\top s \right) \\ &\quad + \left( L\chi - L \sum_{i\in I} k_i e_i^\top \begin{bmatrix} \chi_i \\ y_i \end{bmatrix} \right) (\mathrm{mod} q) \\ &= b\lfloor \frac{q}{2} \rfloor + \underbrace{\left( L\chi - L \sum_{i\in I} k_i e_i^\top \begin{bmatrix} \chi_i \\ y_i \end{bmatrix} \right)}_{\text{error term}} (\mathrm{mod} q). \end{aligned} \tag{12}$$

Let $e_i = \begin{bmatrix} e_{1,i} \\ e_{2,i} \end{bmatrix}$, and $e_{1,i}, e_{2,i} \in \mathbb{Z}^m$. The error term is as follows:

$$\begin{aligned} L\chi - L \sum_{i\in I} k_i e_i^\top \begin{bmatrix} \chi_i \\ y_i \end{bmatrix} &= L\chi - L \sum_{i\in I} k_i e_i^\top \begin{bmatrix} \chi_i \\ R_i\chi_i \end{bmatrix} \\ &= L\chi - L \sum_{i\in I} k_i \left( e_{1,i}^\top + e_{2,i}^\top R_i \right) \chi_i. \end{aligned} \tag{13}$$

From Section 4.2, we know that $\max\{k_i\} = l$ and $(l!)^2 \le (l)^{2l}$.

By Lemma 1, we have $|e_{1,i}^\top|, |e_{2,i}^\top| \le \overline{\sigma}_i \sqrt{m} = \sigma_i m \omega(\sqrt{\log m})$.

By Lemma 3, $\|e_{1,i}^\top + e_{2,i}^\top R_i\| \le \|e_{1,i}^\top| + |e_{2,i}^\top R_i\| \le O(\sigma_i m)$.

Finally, by Lemma 2, $|(e_{1,i}^\top + e_{2,i}^\top R_i)\chi_i|$ is bounded w.h.p. by

$$\begin{aligned} &\left| \left( e_{1,i}^\top + e_{2,i}^\top R_i \right) \chi_i \right| \\ &\le \left\| e_{1,i}^\top + e_{2,i}^\top R_i \right\| q\alpha_i \omega(\sqrt{\log m}) + \left\| e_{1,i}^\top + e_{2,i}^\top R_i \right\| \frac{\sqrt{m}}{2} \\ &\le q\alpha_i \sigma_i m \omega(\log m) + \sigma_i m^{3/2} \omega(\sqrt{\log m}). \end{aligned} \tag{14}$$

Hence, the error term

$$\begin{aligned} &\left| L\chi - L \sum_{i\in I} k_i \left( e_{1,i}^\top + e_{2,i}^\top R_i \right) \chi_i \right| \\ &\le L|\chi| + \left| L \sum_{i\in I} k_i \left( e_{1,i}^\top + e_{2,i}^\top R_i \right) \chi_i \right| \\ &\le (l!)^2 |\chi| + l \cdot (l!)^2 \left| \sum_{i\in I} \left( e_{1,i}^\top + e_{2,i}^\top R_i \right) \chi_i \right| \\ &\le (l)^{2l} \cdot q\alpha_i \omega(\sqrt{\log m}) + (l)^{2l+1} q\alpha_i \sigma_i m \omega(\log m) \\ &\quad + (l)^{2l+1} \sigma_i m^{3/2} \omega(\sqrt{\log m}). \end{aligned} \tag{15}$$

To make the system work correctly, we need the following:

(1) For the *TrapGen* algorithm which can operate, it needs $m \approx 2n \log q$.

(2) For the IMSampleLeft and SampleRight algorithms which can operate, it needs $\sigma_i > \sigma_{TG} \sqrt{m}\omega(\sqrt{\log m})$, where $\sigma_{TG} = O(\sqrt{n \log q})$.

(3) For the error term which is less than $q/5$ w.h.p., it sets

$$\begin{aligned} &\alpha_i < \left[ (l)^{2l} \omega(\sqrt{\log m}) + (l)^{2l+1} \sigma_i m \omega(\log m) \right]^{-1}, \\ &q > (l)^{2l+1} \sigma_i m^{3/2} \omega(\sqrt{\log m}). \end{aligned} \tag{16}$$

(4) For Regev's LWE reduction applied, it needs $q > 2\sqrt{n/\alpha_i}$, $i \in \{1, 2, ..., l\}$.

*4.4. Security*

**Theorem 2.** *Suppose there exists a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ with advantage $\varepsilon > 0$ in a selective security attack against our space-efficient KP-ABE scheme from lattices; then there exists a PPT simulator $\mathcal{B}$ that decides $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem with advantage $\varepsilon/2$.*

*Proof.* In Definition 5, the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem gives access to a sampler $\mathcal{O}$, which is either a truly random sampler $\mathcal{O}_s'$ or a noisy pseudorandom sampler $\mathcal{O}_s$. The decisional algorithm needs to distinguish which the sampler it is given. It proceeds as follows:

*Instance.* $\mathcal{B}$ requests from $\mathcal{O}$ to obtain $(m + 1)$ LWE samples that we denote as

$$\{(w_0, v_0), (w_1, v_1), (w_2, v_2), ..., (w_m, v_m)\} \in \left( \mathbb{Z}_q^n \times \mathbb{Z}_q \right). \tag{17}$$

*Target.* The adversary $\mathcal{A}$ announces to $\mathcal{B}$ the challenge attribute set. Let $A^* = \{1: a_1^*, 2: a_2^*, ..., l: a_l^*\}$ denote the challenge attribute set.

*Setup*. $\mathcal{B}$ constructs the public parameter as follows:

(1) Construct A and u from the LWE instance; let $A = (w_1, w_2, ..., w_m)$ and $u = w_0$. Then the simulator $\mathcal{B}$executes the *TrapGen* algorithm to generate the matrix $B$ with the trapdoor $T_B$.

(2) For each attribute $j \in U$ such that $j \in A^*$, choose $R_j^* \in \{-1, 1\}^{m \times m}$ and compute $A_j = AR_j^* - H(a_i^*)B$.

(3) For each attribute $j \in U$ such that $j \notin A^*$, the simulator $\mathcal{B}$ executes the TrapGen algorithm to generate the matrices $A_j$ with the trapdoor $T_{A_j}$.

(4) $\mathcal{B}$ sends the public parameter to $\mathcal{A}$.

*Queries*. In this step, $\mathcal{A}$ makes queries for the privacy keys adaptively for the policy $(M, \rho)$ that the target attribute set $A^*$ does not satisfy. $\mathcal{B}$ answers the queries as follows:

(1) As in the real scheme, $\mathcal{B}$ construct a low-norm linear sharing matrix $M \in \mathbb{Z}^{l \times n}$.

(2) Let $A\prime = \{1: a_1', 2: a_2', , ..., l: a_l'\}$ denote the set of attribute on choice policy $(M, \rho)$. Note that $A' \neq A^*$.

(3) For each attribute $j \in U$ such that $j \in A'$, construct vectors $g_1, g_2, ..., g_n$ from $u$ like the real scheme. Let $\lambda_j = (M_j g_1, M_j g_2, ..., M_j g_n)^\top = (\lambda_j\ (1), \lambda_j^{(2)}, ..., \lambda_j^{(n)})^\top$.

(4) Compute $H(a'_{\rho(j)})B$ and let $F_{\rho(j)} = A|A_{\rho(j)} + H(a'_{\rho(j)})B$.

(5) Compute $e_{\rho(1)}, e_{\rho(2)}, ..., e_{\rho(l)}$ by using the SampleRight $(A, (H(a^*_{\rho(j)}) - H(a_{\rho(j)}'))B, R, T_B, \lambda_j, \sigma)$ algorithm; satisfy $F_{\rho(j)} e_{\rho(j)} = \lambda_j$.

(6) Finally $\mathcal{B}$ sends the privacy key $e_{\rho(j)}$ to $\mathcal{A}$.

*Challenge*. $\mathcal{A}$ gives a signal that it is ready to accept the challenge. Then it selects a message bit $b^* \in \{0, 1\}$ and sends the message bit to $\mathcal{B}$. The simulator $\mathcal{B}$ responds with a ciphertext $c^* = \left\{c_0^*, (c_j^*)_{j \in A^*}\right\}$ which is encrypted under the target attribute set $A^*$. It executes as follows:

(1) Compute $L = (l!)^2$.

(2) Let $v^* = (v_1, v_2, ..., v_m)^\top$.

(3) The challenge ciphertext is as follows:

$$c_0^* = Lv_0 + \lfloor \frac{q}{2} \rfloor \cdot b^* \in \mathbb{Z}_q,$$

$$c_j^* = L \begin{bmatrix} v^* \\ \left(R_j^*\right)^\top v^* \end{bmatrix} \in \mathbb{Z}_q^{2m}. \tag{18}$$

Note that when the LWE oracle is a pseudorandom sampler $\mathcal{O}_s$, the ciphertext $c = \left\{c_0^*, c_j^*\right\}$ is valid.

As we know $v_0 = w_0^\top s + \chi_0$ and $v^* = (v_1, v_2, ..., v_m)^\top = (w_1^\top s + \chi_1, w_2^\top s + \chi_2, ..., w_m^\top s + \chi_m)^\top = A^\top s + \chi$. Thus, the ciphertext is as follows:

$$c_0^* = Lv_0 + \lfloor \frac{q}{2} \rfloor \cdot b^* = L\left(w_0^\top s + \chi_0\right) + \lfloor \frac{q}{2} \rfloor \cdot b^* = L\left(u^\top s + \chi_0\right) + \lfloor \frac{q}{2} \rfloor \cdot b^*,$$

$$c_j^* = L \begin{bmatrix} v^* \\ \left(R_j^*\right)^\top v^* \end{bmatrix} = L \begin{bmatrix} A^\top s + \chi \\ \left(R_j^*\right)^\top \left(A^\top s + \chi\right) \end{bmatrix} = L \begin{bmatrix} A^\top s + \chi \\ \left(AR_j^*\right)^\top s + \left(R_j^*\right)^\top \chi \end{bmatrix} = L \left[ F_j^\top s + \begin{bmatrix} \chi \\ \left(R_j^*\right)^\top \chi \end{bmatrix} \right]. \tag{19}$$

The ciphertext is encrypted under the target attribute set $A^*$; as we know that $A_j = AR_j^* - H(a_i^*)B$ and $F_j = A|A_j + H(a_j^*)B$, then $F_j = A|AR_j^* - H(a_i^*)B + H(a_j^*)B = A|AR_j^*$.

When $\mathcal{O} = \mathcal{O}_s'$, the ciphertext $c = \left\{c_0^*, c_j^*\right\}$ is uniformly random in $(\mathbb{Z}_q \times \mathbb{Z}_q^m)$.

*Continuation*. After having obtained the challenge ciphertext, $\mathcal{A}$ is allowed to make repeat for the privacy key queries.

*Decision*. $\mathcal{A}$ eventually emits a guess $b'$, whether $c^*$ was actually a valid encryption of $b^*$ as requested. $\mathcal{B}$ uses the guess to determine an answer on the LWE oracle $\mathcal{O}$. If $b' = b^*$, $\mathcal{B}$ answers that the LWE sampler is $\mathcal{O}_s$; otherwise (i.e., $b' \neq b^*$) it is the truly random sampler $\mathcal{O}_s'$.

If the adversary $\mathcal{A}$ succeeds in guessing the message bit (i.e., $b' = b^*$) with probability at least $1/2 + \varepsilon$, then the simulator $\mathcal{B}$ would correctly guess the nature of the LWE oracle with probability at least $1/2 + \varepsilon/2$. □

## 5. Performance Analysis

Here, we give the comparison between our KP-ABE scheme and the related lattice-based ABE scheme in different aspects.

As shown in Table 1, the lattice-based ABE schemes in [8, 15] just support a single AND gate or a single THRESHOLD gate in the attribute matching phase. But our scheme and [16] use the LSSS technique to support three operations of attribute, that is, AND, OR, and THRESHOLD. In addition, the lattice dimension in [8, 15] mostly is $m > 5n \log q$. It will lead to a large trapdoor size. But the lattice dimension in our construction and [16] is approximately equal to $2n \log q$. Meanwhile, [8, 15, 16] introduce the trapdoor generation algorithm of GPV08 to generate a lattice basis as the trapdoor. But in our construction, we introduce the new trapdoor generation algorithm of MP12 [19]. The trapdoor is no longer a lattice basis as [8, 15, 16]. The storage cost of a single trapdoor grows only linearly in

Table 1: The comparing of related schemes.

| Scheme | Access policy | Dimension $m$ | Trapdoor size |
|---|---|---|---|
| Reference [8] | AND | $\geq 6n\log q$ | $m^2$ ( $\geq 36n^2\log^2 q$) |
| Reference [15] | THRESHOLD | $\geq 5n\log q$ | $m^2$ ( $\geq 25n^2\log^2 q$) |
| Reference [16] | AND, OR, and THRESHOLD | $> 2n\log q$ | $m^2$ ( $> 4n^2\log^2 q$) |
| Ours | AND, OR, and THRESHOLD | $\approx 2n\log q$ | $n\log q(m - n\log q)$ ( $\approx n^2\log^2 q$) |

Table 2: The comparison between our scheme and related KP-ABE schemes.

| | Reference [16] | Reference [17] | Ours |
|---|---|---|---|
| $pp$ size | $(\ell + 1)nm\log q + n\log q$ | $(\ell + 2)nm\log q$ | $(s + 2)nm\log q + n\log q$ |
| $msk$ size | $\ell m^2$ | $m^2$ | $\overline{m} \times \overline{n}$ |
| $sk_u$ size | $[(\ell + 1 + \theta)m]^2$ | $2m^2$ | $2m\lvert A_u\rvert$ |
| Ciphertext size | $(\ell + 1)m\log q$ | $(\ell + 2)m\log q$ | $(2\lvert A_c\rvert)m\log q + \log q$ |
| Privacy preserving | No | No | Yes |
| Attribute space | Bounded | Bounded | Unbounded |

$\overline{m} = m - \overline{n}, \overline{n} = nt, t = \lceil \log_2 q \rceil$, and $0 < \theta \leq \ell$. $\ell$: the maximum number of system attributes in [16, 17]. $s$: the maximum number of system attribute labels in our scheme. $|A_u|$: the number of user attributes. $|A_c|$: the number of ciphertext attributes. Note that $A_u \leq s < \ell$ and $A_c \leq s < \ell$.

the lattice dimension $m$ in our construction, rather than quadratically as a basis does in [8, 15, 16]. The trapdoor size is at least four times smaller than the others, even 36 times.

As shown in Table 2, we compare our KP-ABE schemes from lattices with the related lattice-based KP-ABE schemes in storage cost. To make the comparison more clearly, here we let $s$ denote the maximum number of system attribute labels rather than $l$ in our scheme. Note that $l$ in our scheme denotes attribute labels, and $s = \max|l| < \ell$, where $\ell$ is the maximum number of system attributes in [16, 17]. Particularly, due to the fact that we extended the traditional 1-dimensional (value-specified) attribute structures to 2-dimensional (label- and value-specified) ones, the number of system attribute labels is far less than the number of system attributes (i.e., $s < \ell$); the $pp$ size and the $msk$ size are particularly less than the others. Moreover, the $sk_u$ size and the ciphertext size are only correlated to the number of $A_u$ and $A_c$; they actually are lower than others on account of the fact that the size of $sk_u$ and ciphertext in [16, 17] are related to the total number of the system attributes. Besides, in [16, 17], they use a set $\{1, \ldots, \ell\}$ to denote the system attributes and the number of system attributes is fixed in the Setup phase. But in our scheme, the 2-dimensional (label- and value-specified) attribute structure can ensure an unbounded attribute space; that is, it can add new attributes dynamically without reconstructing the system. Attribute values often contain more privacy information than attribute labels. By doing that, attribute label is used to set the access structure, and attribute value is hidden; it can resolve the privacy-preserving problem to some extent. The detailed storage overhead comparison is shown in Figure 1. According to the suggestion given by Micciancio and Peikert in [19], we set the parameter $n = 284$ and $q = 2^{24}$. For the number of the system attributes, according to the suggestion given in [27, 28], we, respectively, let $\ell = 64$, $\ell = 128$, and $\ell = 256$. Since in our scheme, we classify the attributes and assign a label to each attribute. Multiple attribute values may have the same attribute label. Thus, we, respectively, let $s = 8$, $s = 16$, and $s = 32$. It is obvious that the space cost is remarkably reduced due to the 2-dimensional attribute structure and removing the reliance on the total number of system attributes.

As shown in Table 3, we compare our scheme with related lattice-based KP-ABE scheme on time complexity. According to the suggestion given in [19, 27], we let $\ell = n/4$ and $\max|A_c| = l/2$. The encryption time complexity in our construction is equal to [16, 17], that is, $O(n^3 m)$. But the actual encryption time in our construction would be 4 times of [16, 17]. As for the decryption time, in [16] the user's secret key is a $(\ell + 1)m \times (\ell + 1)m$ matrix, while it is a $2m \times m$ matrix in [17]; thus, the decryption time is longer than our construction due to the fact that the user's secret key in our construction is some independent vectors which are related to the number of a user's attributes.

In summary, by introducing a new trapdoor generation algorithm and removing the reliance on the total number of system attributes, our lattice-based KP-ABE scheme solves the master secret key leakage problem. In addition, the 2-dimensional attribute structure enables our scheme to support unbounded attribute space and privacy preserving. The storage cost is remarkably reduced with an acceptable time cost. The flexible access policy makes the scheme in this paper more applicable to the distributed cloud storage environment.

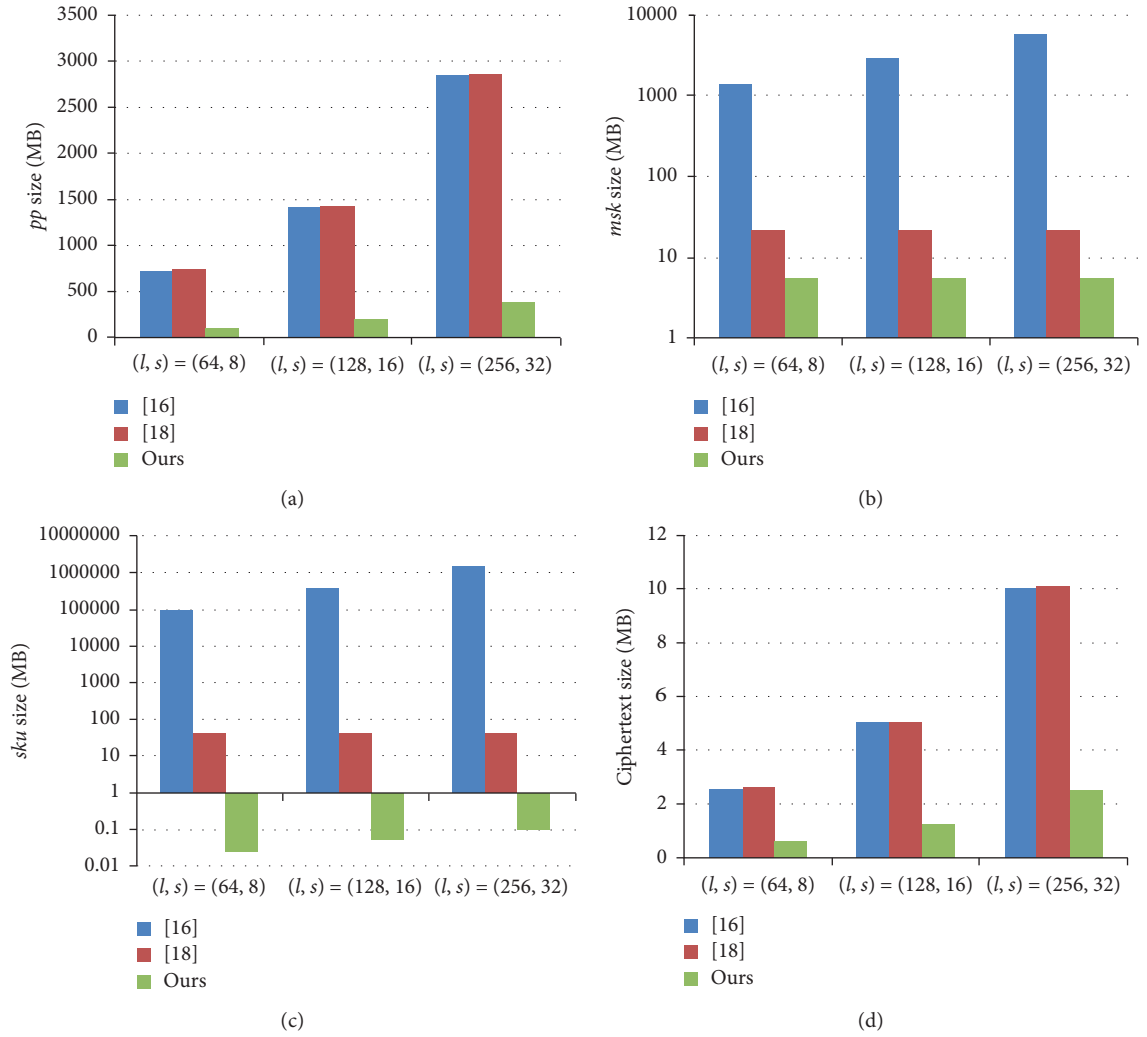


Figure 1: The storage overhead comparing between our scheme and related lattice-based KP-ABE schemes. (a) *pp* size. (b) *msk* size. (c) $sk_u$ size. (d) Ciphertext size.

Table 3: The comparison of time complexity.

| | Reference [16] | Reference [17] | Ours |
|---|---|---|---|
| Encryption | $O(\ell n^2 m)$ | $O(\ell n^2 m)$ | $O(n^3 m)$ |
| Decryption | $O(n\|A_c\|^2 m^2)$ | $O(m^2)$ | $O(\|A_c\|m)$ |

$\ell$: the maximum number of system attributes in [16, 17]. According to [19, 27], let $\ell = n/4$ and $\max|A_c| = l/2$.

## 6. Conclusion

In this paper, based on LSSS technique, we propose a secure KP-ABE scheme from lattice which has solved the divulging problem of the master secret key. In the scheme, we introduced a new trapdoor generation algorithm to generate a strong trapdoor. The *pp* size and *msk* size are all reduced. Moreover, removing the reliance on the total number of system attributes, the $sk_u$ size and the ciphertext size also achieved optimization. Moreover, the description of the attribute is very flexible. Attribute label and attribute value together form an attribute. Thus, it can add new attribute value at any time without rebuilding the system. The attribute space is unbounded. In addition, it also can be extended to a lattice-based large universe multiauthority KP-ABE scheme. Each attribute authority can manage an attribute label and all the values under the attribute label. And how to construct a lattice-based ABE scheme with multiple attribute authorities is our next research direction.

## Data Availability

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

## Conflicts of Interest

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

## Acknowledgments

This work was supported by the National Key R&D Program of China (2017YFB0803001), the Shandong Provincial Key Research and Development Program of China (2018CXGC0701), the National Natural Science Foundation of China (NSFC) (no. 61972050), the BUPT Excellent Ph.D. Students Foundation (nos. CX2019119 and CX2019233), the Team Project of Collaborative Innovation in Universities of Gansu Province (no. 2017-16), and the Major Project of Gansu University of Political Science and Law (no. 2016XZD12).

## References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encrypton," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'05)*, pp. 457–473, Aarhus, Denmark, May 2005.

[2] V. Goyal, O. Pandey, A. Sahai et al., "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 89–98, Alexandria, VA, USA, 2006.

[3] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017.

[4] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'11)*, pp. 568–588, Tallinn, Estonia, May 2011.

[5] Z. Guan, J. Li, L. Zhu, Z. Zhang, X. Du, and M. Guizani, "Toward delay-tolerant flexible data access control for smart grid with renewable energy resources," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3216–3225, 2017.

[6] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2018.

[7] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Information Sciences*, vol. 470, pp. 175–188, 2019.

[8] J. Zhang and Z. F. Zhang, "A ciphertext policy attribute-based encryption scheme without pairing," in *Proceedings of the 7th international conference on information security and cryptology (Inscrypt'11)*, vol. 7537, pp. 324–340, Beijing, China, December 2011.

[9] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual Acm Symposium on Theory of Computing (STOC'05)*, pp. 84–93, Baltimore, MD, USA, May 2005.

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

[11] X. Liu, H. Zhu, Q. Li, T. Zhang, J. Ma, and J. Xiong, "Threshold attribute-based encryption with attribute hierarchy for lattices in the standard model," *IET Information Security*, vol. 8, no. 4, pp. 217–223, 2014.

[12] J. G. Han, W. Susilo, Y. Mu et al., "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 665–678, 2015.

[13] Z. B. Ying, J. F. Ma, and J. T. Cui, "Partially policy hidden CP-ABE supporting dynamic policy updating," *Journal on Communications*, vol. 36, no. 12, pp. 178–189, 2015.

[14] Z. B. Ying, H. Li, J. F. Ma et al., "Adaptively secure ciphertext-policy attribute-based encryption with dynamic policy updating," *Science China Information Sciences*, vol. 59, no. 4, pp. 1–16, 2016.

[15] S. Agrawal, X. Boyen, V. Vaikuntanathan et al., "Functional encryption for threshold functions (or Fuzzy IBE) from lattices," in *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography (PKC'12)*, pp. 280–197, Darmstadt, Germany, May 2012.

[16] X. Boyen, "Attribute-based functional encryption on lattices," in *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography (TCC'13)*, pp. 122–142, Tokyo, Japan, March 2013.

[17] D. Boneh, C. Gentry, S. Gorbunov et al., "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," *Advances in Cryptology-EUROCRYPT 2014*, Springer, pp. 533–556, Berlin, Germany, 2014.

[18] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Attribute-based encryption for circuits," in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC'13)*, pp. 545–554, Palo Alto, CA, USA, June 2013.

[19] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Proceedings of 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'12)*, vol. 7237, pp. 700–718, Cambridge, UK, April 2012.

[20] J. Zhao, H. Y. Gao, and J. Q. Zhang, "Attribute-based encryption for circuits on lattices," *Tsinghua Science and Technology*, vol. 19, no. 5, pp. 463–469, 2014.

[21] Y. T. Wang, "Lattice ciphertext policy attribute-based encryption in the standard model," *International Journal of Network Security*, vol. 16, no. 6, pp. 444–451, 2014.

[22] G. Y. Zhang, J. Qin, and S. Qazi, "Multi-authority attribute-based encryption scheme from lattices," *Journal of Universal Computer Science*, vol. 21, no. 3, pp. 483–501, 2015.

[23] J. Zhao and H. Y. Gao, "LSSS matrix-based attribute-based encryption on lattices," in *Proceedings of the 2017 13th International Conference on Computational Intelligence and Security*, pp. 253–257, Hong Kong, China, December 2017.

[24] L. H. Liu, S. P. Wang, and Q. Yan, "A multi-authority key-policy ABE scheme from lattices in mobile Ad Hoc Network," *Ad-Hoc and Sensor Wireless Networks*, vol. 37, pp. 117–143, 2017.

[25] Y. Liu, L. C. Wang, L. X. Li et al., "Secure and efficient multi-authority attribute-based encryption scheme from lattices," *IEEE Access*, vol. 7, pp. 3665–3674, 2018.

[26] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proceedings of the 30th Annual Conference on Advances in Cryptology (CRYPTO'10)*, pp. 98–115, Santa Barbara, CA, USA, August 2010.

[27] W. Dai, Y. Doröz, Y. Polyakov et al., "Implementation and evaluation of a lattice-based key-policy ABE scheme," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1169–1184, 2018.

[28] K. D. Gür, Y. Polyakov, K. Rohloff, G. W. Ryan, H. Sajjadpour, and E. Savas, "Practical applications of improved Gaussian sampling for trapdoor lattices," *IEEE Transactions on Computers*, vol. 68, no. 4, pp. 570–584, 2019.

[29] 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 (EUROCRYPT'10)*, vol. 6110, Springer, pp. 553–572, Heidelberg, Germany, Heidelberg, Germany, May 2010.

[30] C. Gentry, V. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pp. 197–206, Victoria, Canada, May 2008.