Large Universe Attribute-Based Encryption Scheme from Lattices

Shang-ping Wang*
School of science, Xi’an University of Technology
Xi’an, 710054, Shaanxi, CHINA
spwang99@hotmail.com

Fang Feng
School of science, Xi’an University of Technology
Xi’an, 710054, Shaanxi, CHINA
751770118@qq.com

(Dept of Cryptography and Network Security, Xi’an University of Technology, Xi’an, 710054, China)

Abstract. We propose a large universe attribute-based encryption (ABE) scheme from lattices. It is inspired by Brent Waters’s scheme which is a large universe attribute-based encryption using bilinear map. It is a very practical scheme but this scheme may not be secure with the developing quantum computer. So we extend their good idea of large universe attribute-based encryption to lattices based cryptosystem. And our scheme is the first large universe ABE scheme from lattices. In a large universe ABE system any string can be used as attribute and attributes need not be determined at system setup. This is a desirable feature. And the master private key of our scheme is too short too a matrix. Moreover, our scheme is high efficient due to the ciphertext of our scheme is divided into three parts. Finally, under Learning with Errors (LWE) assumption, we prove our scheme is secure under the select attribute attack.

Key words: Large Universe; Lattices; Attribute-Based Encryption; Short key; Efficiency; Learning with Errors.

1. Introduction

The notion of Fuzzy Identity Based Encryption was first introduced by Sahai and Waters [1] at EUROCRYPT’05, which is the foundation of the emergence of attribute-based encryption (ABE) system. Attribute-Based Encryption has occupied a huge area of research in cryptography over the last few years. The emergence of Attribute-based encryption system makes the decryptor a group with the same properties instead of a single individual. Attribute-based encryption system makes the ciphertext can be shared by multi-user, so it improves access efficiency of the system. The most prominent advantage of the system is very applicable to the situations which not need to know the specific receiver, and the receiver who only satisfies the corresponding conditions can decrypt the message from sender. This makes the attribute-based encryption system be widely used.
The study of attribute based encryption schemes has already obtained many research results. Especially in recent years Waters et al. \cite{papers} made a great improvement of attribute-based encryption schemes in the aspect of practical, the most representative is proposed a scheme which supports large universe attribute encryption. In large universe attribute encryption the size of attribute universe can be exponentially large. This is a desirable feature compared the small universe attribute encryption which fixed attributes at system setup. But it is a pity all this practical scheme is based on bilinear pairing technology with the decisional bilinear Diffie-Hellman assumption or the variant decisional bilinear Diffie-Hellman assumption, such as literature \cite{papers}. But the main drawback of this type of scheme is generally inefficient, because the program is generally necessary to calculate multiple bilinear pairs. So people try to construct the attribute based cryptosystem using other techniques.

In recent years, many breakthroughs have been made in lattice to design cryptosystem, which mainly reflected in the construction of public key cryptosystems relies on lattice hardness problems. Cryptography scheme based on lattice theory has many advantages, such as high efficiency, simple structure which easy to understand. Moreover, the most important advantage is lattice-based schemes enjoying strong security guarantees. There is an intrinsic relation between the attribute based cryptosystem and fuzzy identity based cryptosystem, and there already have some literatures \cite{5,6,7}designed several encryption schemes based on attribute from lattice. Although attribute based scheme on lattice has a little productive.

**Our motivation:** Combined with advantages and disadvantages of ABE from bilinear and the advantages and disadvantages of ABE from lattices, we hope to build a large universe ABE from lattices.

First of all, since Waters et al \cite{papers} propose a practical large universe ABE scheme recently which is a very practical scheme but the security of this scheme is threatened since the emergence of a quantum computer. So we hope to extend his good idea of large universe attribute-based encryption to lattices based cryptosystem.

Second, due to large universe ABE dose not determine the number of attributes in system setup, but we hope determine the master private key at once. This may avoid the flaws of modified the master private key after addition extra attributes later.

Finally, since in large universe ABE scheme, the number of attributes participates in the encryption may too large, so we hope message encryption can divide into different parts. Some parts have nothing to do with the number of attributes participate in the encryption, only one part related. Therefore, we can achieve efficient encryption

**Our Contributions:**

Our main result is the scheme of a large universe attribute-based encryption from lattices, and reduces its security from LWE.

First, our scheme is the first large universe ABE from lattices. We apply new idea access policy \((M, \rho)\) on LSSS which propose in Brent Waters’s \cite{papers} scheme “Practical Constructions and New Proof Methods for Large Universe Attribute-Based Encryption” in 2013 to attribute-based encryption from lattices. And the main idea of this scheme is change the LSSS access policy from one matrix form to a matrix and a \([I] \rightarrow Z_q, \text{map } \rho \text{ form. In previous LSSS access policy, every row of the share-generating matrix } M \text{ is just map to an attribute. While in large }\)
universe ABE, map \( \rho \) in access policy can map every row of the share-generating matrix \( M \) to any attributes on universe. Through use the new access policy \((M, \rho)\) on lattices we get a large universe attribute-based encryption form lattices.

Second, our scheme has a short master key. The master private key of our scheme has only one matrix \( T_{A_0} \), which is a short base of public parameter \( A_0 \). And the public key \( E_i = A_0 \parallel A_i + B \) of every attribute is combined with two parts, the first part is the same public parameter \( A_0 \), and the second part is \( A_i + B \) which is determinant by every different attributes.

So we hide our trapdoor \( T_{A_0} \) information just under the first part. Then the master private key is fixed no matter how many attributes have in our system. But the fault is that the size of ciphertext and the time of encryption will increase two times, since it is linear to the number of columns in \( E_i \).

Third, our scheme is efficient. In order to overcome the shortcoming above, we divide the ciphertext into three parts. And we not use public key \( E_i \) in encryption algorithm directly but divided \( E_i \) into two parts. So our ciphertext combine with three parts: the first part is message encryption, the second part is general encryption and the third part is attributes encryption, where the size of first and second is fixed that is irrelevant to the attributes participant encryption. The first part is only related to the message \( msg \), the second general encryption part just related to the first part of public key \( A_0 \), and the third part is encrypt by the second part \( A_i \) of public key. So will not repeat save and calculation the common part of first part \( A_0 \) of public key. This allows the master private key into a matrix, but the size of the ciphertext and the encrypted time add not too much.

Finally, under LWE assumption, our scheme is security under the select attribute attack.

**Organization:** The rest of this paper is organized as follows. In section 2 we define our notion and recall some preliminaries. In section 3 we describe our construction. The Correctness and Security of our construction are given in section 4. Finally, we conclude in section 5.

### 2. Preliminaries

We first introduce the notion used throughout the paper, and then introduce the basic knowledge of lattice, some theorems on lattice. Finally we give the security assumptions used in our scheme.

#### 2.1 Notation

Throughout this paper, we use uppercase boldface alphabet for matrices, as in \( \mathbf{A} \); lowercase boldface characters for vectors, as in \( \mathbf{e} \), and lowercase regular characters for scalars, as in \( v, \ell \).
be the set \( \{1,2,\cdots,l\} \). Denoted \( l_2 \) norm by \( \| \| \). For \( X \in \mathbb{R}^{n \times m} \) and \( Y \in \mathbb{R}^{n \times m'} \), \( (X \parallel Y) \in \mathbb{R}^{n \times (m+m')} \) denotes the concatenation of the column of \( X \) followed by the columns of \( Y \). Similarly, for \( X \in \mathbb{R}^{n \times m} \) and \( Y \in \mathbb{R}^{n' \times m} \), \( (X;Y) \in \mathbb{R}^{(n+n') \times m} \) is the concatenation of the rows of \( X \) followed by the rows of \( Y \).

### 2.2 Random Integer Lattices

**Definition 1.** Let \( B = [b_1, b_2, \cdots, b_m] \in \mathbb{R}^{n \times m} \) be a \( n \times m \) matrix whose columns are linearly independent vectors \( b_1, b_2, \cdots, b_m \in \mathbb{R}^n \). The \( m \)-dimensional \( \Lambda \) generated by \( B \) is infinite periodic set, \( \Lambda = L(B) = \left\{ y \in \mathbb{R}^m \text{ s.t. } \exists s = (s_1, \cdots, s_m) \in \mathbb{Z}^m, y = Bs = \sum_{i=1}^{m} s_i b_i \right\} \)

Here, we are interested in integer lattices, i.e. infinite periodic subsets of \( \mathbb{Z}^m \).

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

\[
\Lambda_q^+ (A) = \left\{ e \in \mathbb{Z}^m \text{ s.t. } Ae = 0 \pmod{q} \right\}
\]
\[
\Lambda_q^a (A) = \left\{ e \in \mathbb{Z}^m \text{ s.t. } Ae = u \pmod{q} \right\}
\]

### 2.3 Trapdoors for Lattices: The algorithm TrapGen

Ajtai [8] showed how to sample an essentially uniform matrix \( A \in \mathbb{Z}_q^{n \times m} \) with an associated full-rank set \( T_A \subset \Lambda^+ (A) \) of low-norm vectors. We will use an improved version of Ajtai’s basis sampling algorithm due to Alwen and Peikert [9]:

**Definition 3 ([9]).** Let \( n = n(\lambda), q = q(\lambda), m = m(\lambda) \) be positive integers with \( q \geq 2 \) and \( m \geq 5n \log q \). There exists a probabilistic polynomial-time algorithm TrapGen that outputs a pair \( A \in \mathbb{Z}_q^{n \times m}, T_A \in \mathbb{Z}_q^{n \times m} \) such that \( A \) is statistically close to uniform and \( T_A \) is a basis for \( \Lambda^+ (A) \) with length \( L = \| T_A \| \leq m \cdot o \left( \sqrt{\log m} \right) \) with all but \( n^{-o(1)} \) probability.

### 2.4 Discrete Gaussians

**Definition 4.** Let \( m \in \mathbb{Z}_{>0} \) be a positive integer and \( \Lambda \in \mathbb{R}^m \) an \( m \)-dimensional lattice. For any vector \( c \in \mathbb{R}^m \) and any positive parameter \( \sigma \in \mathbb{R}_{>0} \), we define:
\[ \rho_{\sigma,\varepsilon}(x) = \exp \left( -\frac{\|x - c\|^2}{\sigma^2} \right) \]: A Gaussian-shaped function on \( \mathbb{R}^n \) with center \( c \) and parameter \( \sigma \),
\[ \rho_{\sigma,\varepsilon}(\Lambda) = \sum_{x \in \Lambda} \rho_{\sigma,\varepsilon}(x) \]: The (always converging) discrete integral of \( \rho_{\sigma,\varepsilon} \) over the lattice \( \Lambda \),
\[ D_{\Lambda,\sigma,\varepsilon} \): The discrete Gaussian distribution over \( \Lambda \) with center \( c \) and parameter \( \sigma \),
\[ \forall y \in \Lambda, D_{\Lambda,\sigma,\varepsilon}(y) = \frac{\rho_{\sigma,\varepsilon}(y)}{\rho_{\sigma,\varepsilon}(\Lambda)} \]

For notional convenience, \( \rho_{\sigma,0} \) and \( D_{\Lambda,\sigma,0} \) are abbreviated as \( \rho_{\sigma} \) and \( D_{\Lambda,\sigma} \).

2.5 Gaussians Sampling algorithms over Lattices

The following Gaussians Sampling algorithms over lattices are used in basic construction and security proof.

2.5.1 SampleLeft algorithms

SampleLeft algorithms \([10] \): let \( q > 2 \), \( m > n \). There is an algorithm SampleLeft \( (A, B, T_\Lambda, s, u) \) takes a full rank matrix \( A \in \mathbb{Z}_q^{n \times m} \), a matrix \( B \in \mathbb{Z}_q^{m \times m} \), a basis \( T_\Lambda \) of \( \Lambda_q''(A) \), a Gaussian parameter \( s > \|T_\Lambda\| \cdot \omega(\sqrt{\log (m + m_1)}) \) and a vector \( u \) as input, output a vector \( e \in \mathbb{Z}_q^{m + m_0} \) distributed statistically close to \( D_{\Lambda_q''(\Lambda; \sigma)} \), where \( F := (A \parallel B) \).

2.5.2 Preimage Sampling

We will use following algorithm from \([11] \). Let \( q \geq 2 \), \( m \geq 2n \log q \).
Algorithm SamplePre \( (A, T_\Lambda, u, \sigma) \)\([11] \): On input a matrix \( A \in \mathbb{Z}_q^{n \times m} \) with ‘short’ trapdoor basis \( T_\Lambda \) for \( \Lambda_q''(A) \), a target image \( u \in \mathbb{Z}_q^n \) and a Gaussian parameter \( \sigma \geq \|T_\Lambda\| \cdot \omega(\sqrt{\log m}) \), outputs a sample \( e \in \mathbb{Z}_q^n \) from a distribution that is within negligible statistical distance \( D_{\Lambda_q''(\Lambda; \sigma)} \).

2.5.3 Lattice basis extension algorithm

ExtBasis \([12] \): There is a deterministic polynomial-time algorithm ExtBasis \( (S, A' = A \parallel \overline{A}) \) with following properties: given an arbitrary matrix \( A \in \mathbb{Z}_q^{n \times m} \) whose columns generate group \( \mathbb{Z}_q^n \),
an arbitrary basis \( S \in \mathbb{Z}^{m \times m} \) of \( \Lambda^\perp(A) \) and let \( \overline{A} \in \mathbb{Z}^{n \times m} \) be arbitrary.

\[
\text{ExtBasis}(S, A' = A \parallel \overline{A}) \quad \text{outputs a basis } S' \text{ of } \Lambda^\perp(A') \in \mathbb{Z}^{m \times m} \text{ such that } \|S'\| = \|S\|. 
\]

Moreover, the statement holds even if the columns of \( A' \) are permuted arbitrarily (\( A \parallel \overline{A} \) denote the concatenation of matrix \( A \) and \( \overline{A} \)).

### 2.6 Access Structures and Linear Secret-Sharing Schemes

In this section, we present the formal definition of access structures and linear secret-sharing scheme, adapted to match our setting.

**Definition 5 (Access Structures) \([13]\)**. Let \( U \) be the attributes universe. An access structure on \( U \) is a collection \( A \) of non-empty sets of attributes, i.e. \( A \subseteq 2^U \setminus \{\} \). The sets in \( A \) are called the authorized sets and the sets not in \( A \) are called the unauthorized sets.

Additionally, an access structure is called monotone if \( \forall B, C \in U : \text{ if } B \in A \text{ and } B \subseteq C, \text{ then } C \in A \).

In our constructions, we only consider monotone access structures, which mean that a user acquires more attributes; he will not lose his possible decryption privileges.

**Definition 6 (Linear Secret-Sharing Schemes (LSSS)) \([13]\)**. Let \( q \) be a prime and \( U \) the attributes universe. A secret-sharing scheme \( \Pi \) with domain of secrets \( \mathbb{Z}_q \) realizing access structures on \( U \) is linear over \( \mathbb{Z}_q \) if

1. The shares of a secret \( s \in \mathbb{Z}_q \) for each attribute form a vector over \( \mathbb{Z}_q \).

2. For each access structure \( A \) on \( U \), there exists a matrix \( M \in \mathbb{Z}_q^{l \times n} \), called the share-generating matrix, and a function \( \rho \), that labels the rows of \( M \) with attributes from \( U \), i.e. \( \rho : [l] \rightarrow U \), which satisfy the following: During the generation of shares, we consider the column vector \( v = (s, r_2, \ldots, r_n)^T \), where \( r_2, \ldots, r_n \leftarrow Z_q \). Then the vector of \( l \) shares of the secret \( s \) according to \( \Pi \) is equal to \( Mv \in \mathbb{Z}_q^l \). The share \( (Mv)_j \) where \( j \in [l] \) “belong” to attribute \( \rho(j) \).

We will be referring to the pair \( (M, \rho) \) as the policy of the access structure \( A \).

According to \([13]\), each secret-sharing scheme should satisfy the reconstruction requirement for the authorized sets and any unauthorized set cannot reveal any partial information about the
secret. In our setting, let $S$ denote an authorized set for the access structure $A$ encoded by the policy $\left(M, \rho\right)$. Then let $I$ be the set of rows whose labels are in $S$, i.e., $I = \{i \mid i \in \mathbb{I} \land \rho[i] \in S\}$. The reconstruction requirement asserts that the vector $(1, 0, \ldots, 0)$ is in the span of rows of $M$ indexed by $I$. This means that there exist constants $\{w_i\}_{i \in I}$ such that for any valid shares $\{\lambda_i = (Mv)_i\}_{i \in I}$ of a secret $s$ according to $\Pi$, it is true that $\sum_{i \in I} w_i \lambda_i = s$.

Additionally, it has been proved in [13] that the constants $\{w_i\}_{i \in I}$ can be found in time polynomial in the size of the share-generating matrix $M$.

### 2.7 Hardness Assumption

The \textit{LWE} (learning with errors) problem was first defined by Oded Regev [14] in 2005, and has since been extensively studied and used. We use the decisional version of the \textit{LWE} problem.

**Definition 7.** Consider a prime $q$, a positive integer $n$, and a distribution $\chi \in \mathbb{Z}_q^n$, all public. An $\left(Z_q, n, \chi\right)$–\textit{LWE} problem instance consists of access to an unspecified challenge oracle $O$, being, either, a noisy pseudo-random sampler $O_x$ carrying some constant random secret key $x \in \mathbb{Z}_q^n$, or, a truly sampler $O_y$, whose behaviors are respectively as follows:

- $O_x$: Output noisy pseudo-random samples of the form $(w_i, v_i) = \left(w_i, w_i^T x + \chi_i\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$,

where, $x \in \mathbb{Z}_q^n$ is a uniformly distributed persistent secret key that is invariant across invocations.

- $\chi_i$ is a freshly generated ephemeral additive noise component with distribute $\chi$, and $w_i \in \mathbb{Z}_q^n$ is a fresh uniformly distributed vector revealed as part of the output.

- $O_y$: Output truly random samples $(w_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$, drawn in independently uniformly at random in the entire domain $\mathbb{Z}_q^n \times \mathbb{Z}_q^n$.

The $\left(Z_q, n, \chi\right)$–\textit{LWE} problem statement, or \textit{LWE} for short, allows an unspecified number of queries to be make to the challenge oracle $O$, with no stated or prior bound. We say that an algorithm $A$ decides the $\left(Z_q, n, \chi\right)$–\textit{LWE} problem if $\Pr[A^{\chi} = 1] - \Pr[A^{O_y} = 1]$ is non-negligible for a random $x \in \mathbb{Z}_q^n$.

It has been shown in [14] that there is a $\text{poly}(n, q)$–time reduction from
Search \( \left( Z_q, n, \chi \right) - LWE \) to decision \( \left( Z_q, n, \chi \right) - LWE \).

The confidence in the hardness of the \( LWE \) problem stems in part of a result of Regev\[14\] which shows that the for certain noise distributions \( \chi \), the \( LWE \) problem is as hard as the worst-case \( SIVP \) and \( \text{GapSVP} \) under a quantum reduction. A classical reduction with related parameters was later obtained by Peikert\[5\].

**Proposition** (\[14\], Theorem 1). Consider a real parameter \( \alpha = \alpha(n) \in (0,1) \) and a prime \( q = q(n) > 2\sqrt{n}/\alpha \). Denote by \( T = R/Z \) the group of reals \([0,1)\) with addition modulo 1. Denote by \( \Psi_a \) the distribution over \( T \) of a normal variable with mean 0 and standard deviation \( 2\sqrt{n}/\alpha \) then reduced modulo 1. Denote by \( \lfloor x \rfloor = \left\lfloor x + \frac{1}{2} \right\rfloor \) the nearest integer to the real \( x \in R \).

Denote by \( \Psi_a \) the discrete distribution over \( Z_q \) of the random variable \( \left[ qX \right] \mod q \) where the random variable \( X \in T \) has distribution \( \Psi_a \).

Then if there exists an efficient, possibly quantum, algorithm for deciding the \( \left( Z_q, n, \overline{\chi} \right) - LWE \) problem, there exists a quantum \( q \cdot \text{poly}(n) \)-time algorithm for approximating the \( SIVP \) and \( \text{GapSVP} \) problem, to within \( \tilde{O}(n/\alpha) \) factors in the \( l_2 \) norm, in the worst case.

Since the best known algorithms for \( 2^k \)-approximations of \( \text{GapSVP} \) and \( SIVP \) run in time \( 2^\tilde{O}(\sqrt{k}) \)[7], it follows from the above that the \( LWE \) problem with the noise ratio \( \alpha = 2^{-\kappa} \) is likely hard for some constant \( \varepsilon < 1 \).

Next two lemmas will need to show that decryption works correctly.

**Lemma 9**\[15\]. For any n-dimension lattice \( \Lambda \), vector \( \mathbf{e} \in R^n \), and real \( \varepsilon \in (0,1), s > \eta_\varepsilon(\Lambda) \), we have

\[
\Pr_{\mathbf{x} \sim D_{\Lambda, s, \varepsilon}} \left[ \left\| \mathbf{x} - \mathbf{e} \right\| > s\sqrt{n} \right] \leq \frac{1 - \varepsilon}{1 + \varepsilon} \cdot 2^{-n}
\]

The lemma sates that for large enough \( s \), almost the elements chosen from \( D_{\Lambda, s, \varepsilon} \) are close to \( \mathbf{e} \).

**Lemma 10**\[16\]. Let \( \mathbf{e} \) be some vector in \( Z_m \) and let \( \mathbf{y} \sim \overline{\Psi}_a^m \). Then the quantity \( \left\| \mathbf{e}^T \mathbf{y} \right\| \) treated as an integer in \([0, q - 1]\) satisfies
\[ |x^Ty| \leq \|x\|q\alpha \omega(\sqrt{\log m}) + \|x\|\sqrt{m/2} \]

With all but negligible probability in \( m \). In particular, if \( x \leftarrow \mathbb{F}_q \) is treated as an integer in \([0, q - 1]\) then \( |x| \leq q\alpha \omega(\sqrt{\log m}) + \frac{1}{2} \) with all but negligible probability in \( m \).

3. A Large Universe Attribute-Based Encryption Scheme from Lattices

Our scheme is inspired by large universe ABE on bilinear and ciphertext policy ABE from lattices proposed in [2] and [7] respectively. And combine them we achieve our scheme large universe ABE from lattices. The concrete construction consists of the following four PPT algorithms:

**Setup**. \( (1^\lambda) \). Takes the security parameter \( \lambda \) and a description of attributes universe \( U = Z_q \) as input, and outputs the public parameters \( pp \) and the master private key \( msk \). Where prime \( q = \Theta(2^\lambda) \).

1. Use algorithm \( \text{TrapGen}(1^\lambda) \) to select a uniformly random \( n \times m \) matrix \( A_0 \in Z_q^{n \times m} \) with full-rank \( m \) -vector base \( T_{A_0} \in Z_q^{m \times m} \) such that \( T_{A_0} \in \Lambda_q^+(A_0) \) and \( \|T_{A_0}\| \leq m \cdot \omega(\sqrt{\log m}) \);

2. Select a matrix \( B \in Z_q^{n \times m} \) randomly;

3. For each attributes \( i \in U \), select matrix \( A_i \in Z_q^{n \times m} \) randomly;

4. Select a uniform random \( l \)-vector \( s^T = (s_1, s_2, \ldots, s_l) \in Z_q^l \);

5. Return the public parameter \( pp = (A_0, B, \{A_i\}_{i \in U}, s) \) and the master private key \( msk = T_{A_0} \).

**KeyGen** \( (pp, msk, (M, \rho)) \). On input the public parameter \( pp \), the master private key \( msk = T_{A_0} \) and an access policy \( (M, \rho) \), where \( M \in Z_q^{l \times n} \), \( \rho : [l] \rightarrow Z_q \), output the private key \( sk \).

1. Select vector \( y_1, y_2, \ldots, y_l \in Z_q^n \) such that \( y_1 = (s_{11}, y_{12}, y_{13}, \ldots, y_{1n})^T, y_2 = (s_{21}, y_{22}, \ldots, y_{2n})^T, \ldots, y_l = (s_{l1}, y_{l2}, \ldots, y_{ln})^T \), where \( s_1, s_2, \ldots, s_l \in Z_q \) is the corresponding components of \( s^T = (s_1, s_2, \ldots, s_l) \in Z_q^l \), and \( y_{i,j} \in Z_q \) is
choose randomly for \( i = 1, \cdots l; j = 1, 2, \cdots, n \);

2. Let \( M_{y_j} = \left( \hat{\lambda}_1^{(j)}, \hat{\lambda}_2^{(j)}, \cdots, \hat{\lambda}_l^{(j)} \right)^T, i = 1, \cdots l. \)

3. Use SampleLeft algorithm get the private key of attributes \( \rho(i) \), for \( i = 1, \cdots l. \) let

\[
\hat{\lambda}_i^{(j)} = \left( 0, \cdots, 0, \hat{\lambda}_j^{(j)}, 0, \cdots, 0 \right)_{n \uparrow j \uparrow 0}^T \quad \text{and} \quad E_{\rho(i)} = A_0 \| \hat{A}_{\rho(i)} + B, \] 

then call the algorithm \( \text{SampLefte} \left( A_0, \hat{A}_{\rho(i)} + B, \sigma, \hat{\lambda}_i^{(j)} \right) \) get \( \epsilon_{\rho(i)}^{(j)} \in Z_q^{2m+1} \), i.e.

\[
E_{\rho(i)} \epsilon_{\rho(i)}^{(j)} = \left( 0, \cdots, 0, 0, \cdots, 0 \right)_{n \uparrow j \uparrow 0}^T \in Z_q^n \quad \text{where} \quad \hat{\lambda}_i^{(j)} = \left( 0, \cdots, 0, \hat{\lambda}_j^{(j)}, 0, \cdots, 0 \right)_{n \uparrow j \uparrow 0}^T, \text{namingly}
\]

\[
E_{\rho(i)} \epsilon_{\rho(i)}^{(j)} = \left( \hat{\lambda}_1^{(j)}, \cdots, 0 \right)^T; E_{\rho(i)} \epsilon_{\rho(i)}^{(2j)} = \left( 0, \hat{\lambda}_2^{(j)}, \cdots, 0 \right)^T; \cdots; E_{\rho(i)} \epsilon_{\rho(i)}^{(j)} = \left( 0, 0, \cdots, \hat{\lambda}_j^{(j)} \right)^T \\
E_{\rho(2)} \epsilon_{\rho(2)}^{(j)} = \left( \hat{\lambda}_1^{(j)}, \cdots, 0 \right)^T; E_{\rho(2)} \epsilon_{\rho(2)}^{(2j)} = \left( 0, \hat{\lambda}_2^{(j)}, \cdots, 0 \right)^T; \cdots; E_{\rho(2)} \epsilon_{\rho(2)}^{(j)} = \left( 0, 0, \cdots, \hat{\lambda}_j^{(j)} \right)^T \\
\vdots
\]

\[
E_{\rho(i)} \epsilon_{\rho(i)}^{(j)} = \left( \hat{\lambda}_1^{(j)}, \cdots, 0 \right)^T; E_{\rho(i)} \epsilon_{\rho(i)}^{(2j)} = \left( 0, \hat{\lambda}_2^{(j)}, \cdots, 0 \right)^T; \cdots; E_{\rho(i)} \epsilon_{\rho(i)}^{(j)} = \left( 0, 0, \cdots, \hat{\lambda}_j^{(j)} \right)^T 
\]

Where \( \sigma \geq m \cdot \omega(\log m) \geq \| T_{\lambda_0} \| \omega \left( \sqrt{\log m} \right) \), then according to the Algorithm \( \text{SampLefte} \left( A_0, \hat{A}_{\rho(i)} + B, \sigma, \hat{\lambda}_i^{(j)} \right) \) we know \( \epsilon_{\rho(i)}^{(j)} \) from a distribution which close to the distribution \( D_{\hat{\lambda}_i^{(j)} \left( \lambda_0, \sigma \right)} \). So we know \( \| \epsilon_{\rho(i)}^{(j)} \| \leq \sigma \sqrt{m} \) with overwhelming probability.

4. Obtain the private key \( sk_{\rho(i)} = \left( \epsilon_{\rho(i)}^{(j)}, \epsilon_{\rho(i)}^{(2j)}, \cdots, \epsilon_{\rho(i)}^{(j)} \right) \) of attributes \( \rho \left( i \right) \), and set the private key \( sk_{\rho(i)} = \left( \epsilon_{\rho(i)}^{(j)}, \epsilon_{\rho(i)}^{(2j)}, \cdots, \epsilon_{\rho(i)}^{(j)} \right) \) to the entity who has the attributes \( \rho \left( i \right) \) under access policy \( (M, \rho) \) through secure channel.

Note: Under a access policy \( (M, \rho) \), the decrypt key of an attributes list \( S \) is

\[
\text{sk} = \left( (M, \rho), \left( sk_{\rho(i)} \right)_{i \in [S]} \right).
\]

\( \text{Encrypt} \left( pp, S, \text{msg} \right) \): On input the public parameter \( pp \), an attributes set \( S \subseteq U = Z_p \) and a plaintext \( \text{msg} \in \{0, 1\} \), output ciphertext \( ct \).

1. Choose a uniformly random \( n \)-vector \( \mathbf{x} \in Z_q^n \).
2. Constructs a $n$ dimension vector $f$, just expands $l$ dimension vector $s^T = (s_1, s_2, \cdots, s_l) \in \mathbb{Z}_q^l$ in the public parameter $pp$ to $n$ dimension.

Set $f_i = s_i, i \leq l; f_i = 0, l < i \leq n$;

3. Select a low-norm Gaussian noise scalar $\chi_0 \in \mathbb{Z}_q$ according to some parametric distribution $\mathcal{N}_a$, and select a low-norm Gaussian noise vector $\chi_i, \chi_i \in \mathbb{Z}_q^{2^n}$ from $\left(\mathcal{N}_a\right)^n$ for every $i \in S$; And compute:

\[
C_0 = x^T f + \chi_0 + \left\lfloor \frac{q}{2} \right\rfloor \text{msg} \in \mathbb{Z}_q
\]

\[
C' = x^T A_0 + \chi' \in \mathbb{Z}_q^{2^n}
\]

\[
C_i = x^T (A_i + B) + \chi_i^T \in \mathbb{Z}_q^{2^n}
\]

Then output the ciphertext $ct = (S, C_0, C', \{C_i\}_{i \in S})$.

Decrypt $(pp, sk, ct)$: On input the public parameter $pp$, the private key $sk$ about an attributes set and a ciphertext $ct$. Output the message $\text{msg}$.

First the decryption algorithm calculates the set of rows in $M$ that provide a share to attributes in $S$, i.e. $I = \{i : \rho(i) \in S\}$. Then it computes the constants $\{\omega_i \in \mathbb{Z}_q\}_{i=1}^l$ such that $\sum_{i=1}^l \omega_i M_i = (1, 0, \cdots, 0)$, where $M_i$ is $i$-th row of the matrix $M$. These constants exist if the set $S$ is an authorized set of the policy (see [13]).

Then it calculates $r = C_0 - \sum_{j=1}^l \sum_{i=1}^l \omega_i (C'_i, C_j)e^{(j)}_{\rho(i)}$, if $|r| < \frac{q}{4}$, output 0, else output 1.

### 4. Correctness and Security

#### 4.1 Correctness

In this subsection, we show that our construction is correct with some appropriate parameters setting.

**Theorem 11** if we select the appropriate parameter, the Attribute-Based Encryption Scheme from Lattices in the previous section will be decryption successfully with overwhelming probability.

**Proof** If the decryptor can calculate $\sum_{i=1}^l \omega_i M_i = (1, 0, \cdots, 0)$, and know the private key $sk = \left(\{M, \rho\}, \{sk_{\rho(i)}\}_{i \in S}\right)$, then decrypted by the calculation:
\[
(C'; C_j) = \begin{pmatrix} C' \\ C_j \end{pmatrix} = \begin{pmatrix} x^T A_0 + \chi'^T \\ x^T (A_i + B) + \chi_i^T \end{pmatrix} = \begin{pmatrix} x^T A_0 \\ x^T (A_i + B) \end{pmatrix} + \begin{pmatrix} \chi'^T \\ \chi_i^T \end{pmatrix} = x^T (A_0 \parallel (A_i + B)) + \begin{pmatrix} \chi'^T \\ \chi_i^T \end{pmatrix}
\]

And set \( \chi = (\chi'^T; \chi_i^T) \)

\[
r = C_0 - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (C' ; C_j) e_{r(i)}^{(j)}
= C_0 - \sum_{j=1}^{l} \sum_{i \in I} \omega_i \left( x^T (A_0 \parallel (A_i + B)) + \chi \right) e_{r(i)}^{(j)}
= C_0 - \sum_{j=1}^{l} \sum_{i \in I} \omega_i \left( x^T (A_0 \parallel (A_i + B)) \right) e_{r(i)}^{(j)} - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
= C_0 - x^T \sum_{j=1}^{l} \sum_{i \in I} \omega_i E_j e_{r(i)}^{(j)} - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
= C_0 - x^T \sum_{j=1}^{l} \sum_{i \in I} \omega_i \begin{pmatrix} 0,\ldots,0, \lambda^{(j)}_i, 0,\ldots,0 \\ \lambda^{(j)}_i \\ \lambda^{(j)}_i \end{pmatrix}^T - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
= C_0 - x^T \sum_{j=1}^{l} \sum_{i \in I} \omega_i \begin{pmatrix} 0,\ldots,0, \lambda^{(j)}_i, 0,\ldots,0 \\ \lambda^{(j)}_i \\ \lambda^{(j)}_i \end{pmatrix}^T - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
= C_0 - x^T \sum_{j=1}^{l} \sum_{i \in I} \omega_i \begin{pmatrix} 0,\ldots,0, \lambda^{(j)}_i, 0,\ldots,0 \\ \lambda^{(j)}_i \\ \lambda^{(j)}_i \end{pmatrix}^T - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
= C_0 - x^T \mathbf{f} - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
= \left( x^T \mathbf{f} + \chi_0 + \frac{q}{2} \mathbf{msg} \right) - x^T \mathbf{f} - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
= \frac{q}{2} \mathbf{msg} + \chi_0 - \sum_{j=1}^{l} \sum_{i \in I} \omega_i (\chi) e_{r(i)}^{(j)}
\]

where (1), (2), (3) we use the fact 

\[
E_{r(i)} e_{r(i)}^{(j)} = \begin{pmatrix} 0,\ldots,0, \lambda^{(j)}_i, 0,\ldots,0 \\ \lambda^{(j)}_i \\ \lambda^{(j)}_i \end{pmatrix}^T \in Z_q^n , \sum_{j=1}^{l} \omega_i M_j = (1,0,\ldots,0) \ , \ \text{in fact because}
\]

\[
M y_i = \begin{pmatrix} \lambda^{(j)}_1, \lambda^{(j)}_2, \ldots, \lambda^{(j)}_l \end{pmatrix}^T , \ y_i = (s_i, y_{i_2}, \ldots, y_{i_n})^T \text{ then}
\]
\[ \sum_{i=1}^{\omega_i} \left( \begin{array}{c} 0, \ldots, 0, \alpha_i (j), 0, \ldots, 0 \\ j-1 \rightarrow 0 \\ n-j \rightarrow 0 \end{array} \right) \]

\[ = \sum_{i=1}^{\omega_i} \left( \begin{array}{c} 0, \ldots, 0, M_{y_j}, 0, \ldots, 0 \\ j-1 \rightarrow 0 \\ n-j \rightarrow 0 \end{array} \right) \]

\[ = \left( \begin{array}{c} 0, \ldots, 0, \sum_{i=1}^{\omega_i} M_{y_j}, 0, \ldots, 0 \\ j-1 \rightarrow 0 \\ n-j \rightarrow 0 \end{array} \right) \]

\[ = \left( \begin{array}{c} 0, \ldots, 0, s_j, 0, \ldots, 0 \\ j-1 \rightarrow 0 \\ n-j \rightarrow 0 \end{array} \right) \]

### 4.2 Security

In this subsection, we prove our scheme’s security in the standard model under above parameters setting.

**Theorem 12** If there is a probabilistic polynomial-time algorithm \( \mathcal{A} \) with advantage \( \varepsilon > 0 \) in a selective-security attribute list attack against the above scheme, then there exists a probabilistic polynomial-time algorithm \( \mathcal{B} \) that decides the \( LWE_{n,\alpha} \) problem with advantage \( \varepsilon \), where \( \alpha = O\left(\text{poly}(n)\right) \).

**Proof.** In the \( LWE \) problem, the decision algorithm is given access to a sampling oracle, \( O \), which is either a pseudo-random \( O_x \) with embedded secret \( x \in Z_q^n \), or a truly random sampler \( O_\beta \) (see definition 7). Our decide algorithm \( \mathcal{B} \) will simulate an attack environment for, and exploit the prowess of \( \mathcal{A} \), to decide which oracle it is given. The reduction proceeds as follows.

**Target.** \( \mathcal{A} \) announces a target attribute set \( S^* = \{\text{Attrib}_1, \text{Attrib}_2, \cdots, \text{Attrib}_t\} \subseteq U \), on which it wishes to be challenged.

**Setup.** Challenger \( \mathcal{B} \) get target attribute set from adversary \( \mathcal{A} \).

1. \( \mathcal{B} \) draws \( (s, v) \in Z_q^n \times Z_q \), \( (A_0, v_0) \in Z_q^{n \times m} \times Z_q^n \) from \( O \). And compute \( (B, T_B) \leftarrow \text{TrapGen}\left(n, m, q\right) \).

2. For each attribute \( i \in S^* \), \( \mathcal{B} \) draws \( (A_i, v_i) \in Z_q^{n \times m} \times Z_q^n \) from \( O \), and set \( A_i = A_i - B \) in the public parameter;

3. For each attribute \( i \notin S^* \), challenger \( \mathcal{B} \) draws \( (A_i, v_i) \in Z_q^{n \times m} \times Z_q^n \) from \( O \), and set \( A_i = A_i \) in the public parameter;
Finally, challenger $\mathcal{E}$ set the public parameter $pp = (A_0, B, \{A_i\}_{i \in U}, s)$ and keep $msk = (T_B, v_0, v_0', \{v_i\}_{i \in U})$ as secret.

**Key generation Queries1.** Now the challenger $\mathcal{E}$ has to produce a private key about the access policy $(M, \rho)$, but for this access policy $(M, \rho)$, the target attribute set $S^* = \{\text{Attrib}_1, \text{Attrib}_2, \cdots, \text{Attrib}_j\} \subseteq U$ is a not authorized set. Here we describe the process that the challenger $\mathcal{E}$ how to construct a private key about $(M, \rho)$ as follow (where $M \in Z_q^{\times n}, \rho: [l] \rightarrow Z_q$):

1. Since $S^*$ is not authorized for $(M, \rho)$, there exists an attribute $\rho(\tau) \not\in S^*$ for all $\tau \in [l]$ at least. Without loss of generality, we suppose $\rho(1) \not\in S^*$. By setup algorithm know, the challenger $B$ know the trapdoor $T_B$ of $B$. Then the challenger $\mathcal{E}$ can calculate the trapdoor $T_{E_{\rho(\tau)}}$ of $E_{\rho(\tau)} = A_0 || (A_{\rho(\tau)} + B)$ through algorithm $ExtBasis$.

2. The according the algorithm $ExtBasis(T_{E_{\rho(\tau)}}, E = E_{\rho(1)} || E_{\rho(2)} || \cdots || E_{\rho(l)})$ output the trapdoor $T_E$ of $E$. Then the challenger $\mathcal{E}$ use trapdoor $T_E$ to get private key $sk_{\rho(\tau)}$ of attribute $\rho(k), k \in [l]$ through calling the algorithm

$$\text{SamplePre}(E, T_E, s_j, \sigma)$$ (where $s_j = \left[ \begin{array}{c} 0,0,\ldots,0, s_j, 0, \ldots, 0 \\ j-1 \end{array} \right]$). Namely

$$E \cdot sk_{\rho(k)} = \left[ \begin{array}{c} 0 \\ \vdots \\ s_k \\ \vdots \\ 0 \end{array} \right]_{n \times 1}$$

**Challenge.** When adversary $\mathcal{A}$ submits $msg_0, msg_1 \in \{0,1\}$, challenger $\mathcal{E}$ randomly chooses $b \in \{0,1\}$, and computes the ciphertext about attributes set $S^*$:

$$C_0 = v_0 + \left\lfloor \frac{q}{2} \right\rfloor msg_b \in Z_q$$

$$C' = v_0 \in Z_q^m$$
\[ C_i^* = \mathbf{v}_i \in \mathbb{Z}_q^n, i \in S^* \]

And return ciphertext \[ C^* = (C_0, C', \{ C_i^* \}_{i \in S^*}) \]

**Key generation Queries**. \( \mathcal{E} \) answers queries of \( \mathcal{A} \) the same way it does in key generation queries with the only restriction that the target attributes \( S^* \) is not authorized for \((M, \rho)\) queries.

**Guess.** After the query 2, adversary \( \mathcal{A} \) output a bit \( b' \) as a guess for \( b \). If \( b = b' \), \( \mathcal{E} \) outputs 1, else outputs 0.

**4.3 Parameter**

Because our scheme refer some classical algorithms and consider the correct and security of our scheme, we want the set the parameter appropriately.

1. Use \( \text{TrapGen} \) (see section 2.3 \( \text{TrapGen} \)) in setup, and the algorithm need \( m \geq 5n \log q \);

2. Use \( \text{SamplLefte} \) (See section 2.5.5) in key generation, and need Gaussian parameter \( \sigma > \| \tilde{T}_{\lambda_0} \| \cdot \omega \left( \sqrt{\log 2m} \right); \)

3. For the hardness of \( \text{LWE} \) (see Proposition 8) assumption, and need \( \sigma q \geq 2 \sqrt{m} \); \n
4. For the \( \text{SamplePre} \) (see section 2.5.2) used in key generation queries of security proof, and need \( \sigma \geq \| \tilde{T}_x \| \cdot \omega \left( \sqrt{\log lm} \right). \)

For the correctness of our scheme we need \( \chi_0 - \sum_{j \in \mathcal{L}} \sum_{i=1}^l \omega_i(\chi)e_i^{(j)} < q/2 \). Combine the restriction about parameter above, we set the parameter in our scheme as follows:

Note that according to the \( \text{TrapGen} \) algorithm we know \( \| \tilde{T}_{\lambda_0} \| \cdot \| \tilde{T}_x \| \leq O \left( \sqrt{n \log q} \right) \), and then we calculate \( \chi_0 - \sum_{j \in \mathcal{L}} \sum_{i=1}^l \omega_i(\chi)e_i^{(j)} \).
\[
\left| x_0 - \sum_{j=1}^t \left( \sum_{i=1}^n \omega_i(x) e_i^{(j)} \right) \right| \\
\leq \left| x_0 + \sum_{j=1}^t \left( \sum_{i=1}^n \omega_i(x) e_i^{(j)} \right) \right| \\
\leq \left| x_0 + \sum_{j=1}^t \left( \sum_{i=1}^n \omega_i(x) e_i^{(j)} \right) \right| \\
\leq \left| x_0 + l^2 |\text{e}_i^{(j)}| \right| \\
\leq q\alpha \omega \left( \sqrt{\log 2m} + \frac{1}{2} \right) \left( \left\| \text{e}_i^{(j)} \right\| q\alpha \omega \left( \sqrt{\log 2m} + \sqrt{2m}/2 \right) \right) \\
\leq \left( q\alpha \omega \left( \sqrt{\log 2m} + \frac{1}{2} \right) + l^2 \sqrt{2m} \left( q\alpha \omega \left( \sqrt{\log 2m} + \sqrt{2m}/2 \right) \right) \right) \\
\leq \left( q\alpha \omega \left( \sqrt{\log 2m} + \sqrt{2m}/2 \right) \right) \left( 1 + l^2 \sqrt{2m} \right) \\
\leq \left( q\alpha \omega \left( \sqrt{\log 2m} + \frac{q\alpha}{4} \right) \right) \left( 1 + l^2 \sqrt{2m} \right) \\
= q\alpha \omega \left( \sqrt{\log 2m} + 1 \right) \left( 1 + l^2 \sqrt{2m} \right)
\]

So we set the parameter such that \( q\alpha \left( \omega \left( \sqrt{\log 2m} \right) + 1 \right) \left( 1 + l^2 \sqrt{2m} \right) \leq \frac{q}{5} \) and satisfy four restrictions above, our scheme will be success with overwhelming probability.

Assume that \( \delta \) is a real such that \( n^{1+\delta} > n \log q \), and set \( n, m, q, \alpha \) as follow:

1. Take \( n \) such that \( n^{1+\delta} > n \log q \) appropriate;

2. Set \( m = n^{1.5} \geq 5n \log q \), satisfying 1 above;

3. \( \sigma = ln \cdot \omega \left( \log ln \right) \), satisfying 2 and 4 above;

4. The noise parameter \( \alpha = \frac{1}{5} \left( \left( \omega \left( \sqrt{\log 2m} \right) + 1 \right) \left( 1 + l^2 \sigma \sqrt{2m} \right) \right)^{-1} \) satisfying 3 above;

5. Set large prime \( q \) as \( q \geq 10 \sqrt{2m} \left( \omega \left( \sqrt{\log 2m} \right) + 1 \right) \left( 1 + l^2 \sigma \sqrt{2m} \right) \), satisfying the inequation.

For this setting, it is not only satisfying the condition algorithm we used but also can decrypt ciphertext correctly with overwhelming probability.

5. Conclusions

In this paper, we present a large universe attribute-based encryption from lattices, and the size of the attribute sets can be exponentially large, and we can add attributes after the system setup. The scheme only uses a matrix as the master private key, it makes the master private key’s storage cost is too small; moreover, the ciphertext divide into three parts in the encryption process, and to
do so both reduce the size of the ciphertext and shorten the time encryption; Finally, under LWE assumption, our scheme is security under the select attribute attack.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grants 61173192 and 61303223, Research Foundation of Education Department of Shaanxi Province of China under grants 12JK0740. Thanks also go to the anonymous reviewers for their useful comments.

Reference

[1] Sahai A, Waters B. Fuzzy identity-based encryption [M] Advances in Cryptology EUROCRYPT 2005. Springer Berlin Heidelberg, 2005: 457-473.
[2] Rousselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption[C] Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013: 463-474.
[3] Lewko A, Waters B. Unbounded HIBE and attribute-based encryption [M] Advances in Cryptology EUROCRYPT 2011. Springer Berlin Heidelberg, 2011: 547-567.
[4] Okamoto T, Takashima K. Fully secure unbounded inner-product and attribute-based encryption [M] Advances in Cryptology–ASIACRYPT 2012. Springer Berlin Heidelberg, 2012: 349-366.
[5] Boyen X. Attribute-based functional encryption on lattices [M] Theory of Cryptography. Springer Berlin Heidelberg, 2013: 122-142.
[6] Microsoft Corporation. Attribute based encryption using lattices. US patent application US20120155635, 17 December 2010.
[7] Jiang Zhang, Zhenfeng and Zhang Aijun Ge. Ciphertext Policy Attribute-Based Encryption from Lattices. Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, 2012 Pages: 16-17
[8] Miklos Ajtai. Generating hard instances of the short basis problem. In Jir Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, ICALP, volume 1644 of Lecture Notes in Computer Science, pages 1-9. Springer, 1999.
[9] Joel Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In STACS, pages 75-86, 2009.
[10] Agrawal S, Boneh D, Boyen X. Efficient lattice (H) IBE in the standard model [M] Advances in Cryptology–EUROCRYPT 2010. Springer Berlin Heidelberg, 2010: 553-572.
[11] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, STOC, pages 197-206, ACM, 2008.
[12] David Cash and Dennis Hofheinz. Bonsai Trees, or How to Delegate a Lattice Basis. Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 523-552.ACM,2010.
[13] Beimel A. Secure schemes for secret sharing and key distribution [D]. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
[14] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In
STOC ’05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, pages 84-93, New York, NY, USA, 2005. ACM.

[15] Ajtai M, Kumar R, Sivakumar D. A sieve algorithm for the shortest lattice vector problem[C] Proceedings of the thirty-third annual ACM symposium on Theory of computing. ACM, 2001: 601-610.

[16] Liang X, Cao Z, Lin H, et al. Provably secure and efficient bounded ciphertext policy attribute based encryption [C] Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. ACM, 2009: 343-352.