Group Oriented Attribute-based Encryption Scheme from Lattices with Shamir's Secret Sharing scheme

Maharage Nisansala Sevwandi Perera[1,a)] Toru Nakamura[2,b)]
Takashi Matsunaka[1,c)] Hiroyuki Yokoyama[1,d)] Kouichi Sakurai[1,3,e)]

Abstract: This paper presents the lattice-based construction of group-oriented attribute-based encryption (GO-ABE). The GO-ABE scheme facilitates users from the same group to pool their attributes to match a given ciphertext's decryption policy while securing their associated private keys. This scheme is applicable when no single user can read the message alone, but a set of users can satisfy the decryption policy together. The idea of GO-ABE was first presented by Li et al. in NSS 2015. However, their scheme is not secure against quantum attacks as their scheme's construction is based on bilinear mappings. Ensuring the scheme's security against quantum computers, we construct the scheme using the post-quantum cryptographic primitive lattices and employ Shamir's secret sharing scheme to satisfy GO-ABE requirements.

Keywords: attribute-based encryption, group-oriented, privacy, lattice-based cryptography

1. Introduction

The traditional public key encryption (PKE) schemes enable the sender to encrypt his message targeting a specific recipient only who can decrypt the ciphertext. The traditional PKE schemes are suitable when the identity of the recipient is known by the sender. In 2005, Sahai and Waters [16] proposed the first Attribute-based Encryption (ABE) scheme in which the user secret keys (private keys) and ciphertexts are associated with a set of attributes. A user is allowed to decrypt the ciphertext if and only if there is a match between the attributes of the ciphertext and his secret key. The ABE scheme of Sahai and Waters [16] allows the user to access the message if he can satisfy at least t attributes, where t is the threshold value. For instance, if Alice encrypts a document to the attribute set {A, B, C} and if the threshold value $t = 2$, then Bob with attributes {A, B} can decrypt the document. Later, cryptographers put forth two kinds of ABE schemes depending on whether the access policy is associated with the private key or ciphertext. These two ABE scheme types are Key-Policy Attribute-based Encryption (KP-ABE) schemes and Ciphertext-Policy Attribute-based Encryption (CP-ABE) schemes.

In KP-ABE schemes [1], [14], [17], a ciphertext is associated with a set of attributes and a user private key is associated with an access policy. In contrast, in CP-ABE schemes [4], [6], [7], [11], [12], [19], a private key is associated with a set of attributes, and ciphertext is associated with an access policy. In both settings, decryption is succeeded if and only if there are attributes to match the access policy. KP-ABE is employed in applications to control the data that a user can access. Those applications include purchased (subscribe) broadcasting, structured organizations, and secure forensic analysis. For instance, a user can access only the channels that he purchased in a broadcasting system. The concept of CP-ABE is closer to the traditional access control methods. CP-ABE is employed to control the users. Thus applications including Personal Health Record (PHR) systems employ CP-ABE schemes to control accessing data stored in a cloud system.

PHR is the electronic record of patients. PHR ensures that different clinics and hospitals can access and share the data of a patient. The application of ABE in PHR ensures the security of data. ABE ensures that only a user with certain attributes can access the patient information in the cloud. For instance, only a cardiologist may access the previous cardiology data of the patient Alice. However, in case Alice has a problem with her heart and stomach unless there is a user with attributes {Cardiologist, Gastroenterologist}, the patient data cannot be accessed. Such situations may

[1] Adaptive Communications Research Laboratories, Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan
[2] KDDI Research, Inc., Saitama, Japan
[3] Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan
[a)] perera.nisansala@atr.jp
[b)] tr-nakamura@kddi.com
[c)] ta-matsunaka@atr.jp
[d)] hr-yokoyama@atr.jp
[e)] sakurai@inf.kyushu-u.ac.jp

put the life of the patient in danger. Ming Li et al. [13] proposed beak-glass access to PHRs for emergency scenarios. In simple, when an emergency happens, the staff needs to contact the emergency department (ED) which has the authority to issue temporary decryption keys. After the emergency is over, the patient can revoke the emergent access by contacting ED. However, Li et al. [9] showed that the framework of Ming Li et al. [13] may put the life of the patient in danger if the staff is unable to contact ED. Then they suggested a more flexible mechanism to manage such situations proposing Group Oriented Attribute-base Encryption (GO-ABE) idea.

GO-ABE allows the users from the same group to pool their possessing attributes to match the access policy. For instance, in a PHR system where the access policy requires attributes {Cardiologist, Gastroenterologist}, two doctors Cardiologist and Gastroenterologist together can access the patient's data. However, Li et al. constructed their scheme using bilinear mappings which is not quantum secure.

Contribution

The GO-ABE scheme presented by Li et al. [9] is not quantum-safe. We present the quantum-safe construction for the GO-ABE scheme from lattice cryptography. Li et al. [9] insist that the attribute pooling users are from the same group and they will not reveal their private keys. In other words, comparing to the traditional ABE schemes, GO-ABE requires the users who are pooling the attributes to be from the same group. On the other hand, the users should keep their attributes secured. To satisfy these requirements we employ the lattice-based construction of the Fuzzy IBE scheme presented by Agrawal et al. [3]. Thus as like in Agrawal et al. work, our scheme construction consists of Shamir's Secret Sharing (SSS) scheme and Lagrange interpolation formula. The employment of the SSS scheme and Lagrange formula ensures that the attribute pooling users are from the same group. At the key generation step ℓ shares of a public key $\mathbf{u} = (u_1, \ldots, u_n)$ is constructed using SSS scheme such that the $j-th$ share vector $\hat{\mathbf{u}}_j = (\hat{u}_{j,1}, \ldots, \hat{u}_{j,n})$. The fractional Lagrangian coefficient L_j is calculated such that $\mathbf{u} = \sum_{j \in J} L_j$ where $J \subset [\ell]$. We take the universal attribute size as ℓ. Thus each user gets a secret key for each possessing attribute depending on the group. As a result users from the same group can contribute together to compute $\mathbf{u}$.In other words, no users from different groups can pool together. Our lattice-based construction of GO-ABE is secured in the selective security model under the hardness assumption of learning with errors (LWE) problem.

2. Preliminaries

In this section we provide the notations we use in this paper and provide definitions of lattices with the related algorithms. Moreover we give the syntax of the traditional attribute-based encryption (ABE) scheme.

2.1 Notation

We denote matrices by upper-case bold letters and vectors by lower-case bold letters. For any integer $k \geq 1$, a set of integers $\{1, 2, \ldots, k\}$ is denoted by $[k]$. If S is a finite set, $|S|$ is its size. $S(k)$ indicates its permutations of k elements and $b \leftarrow D$ denotes that b is sampled from a uniformly random distribution D. The encoding function with full rank differences (FRD) $\mathcal{H} : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^{n \times n}$ is taken as discussed by Agrawal et al. [2] paper.

2.2 Lattices

Let q be a prime and $\mathbf{B} = [\mathbf{b}_1 | \cdots | \mathbf{b}_m] \in \mathbb{Z}_q^{r \times m}$ be linearly independent vectors in $\mathbb{Z}_q^r$. The r-dimensional lattice $\Lambda(\mathbf{B})$ for $\mathbf{B}$ is defined as

$$\Lambda(\mathbf{B}) = \{\mathbf{y} \in \mathbb{Z}^r \mid \mathbf{y} \equiv \mathbf{B}\mathbf{x} \bmod q \text{ for some } \mathbf{x} \in \mathbb{Z}_q^m\},$$

which is the set of all linear combinations of columns of $\mathbf{B}$. The value m is the rank of $\mathbf{B}$.

Lattice- based cryptography is quantum resist because the computational problems on lattices believed to be hard to solve, even for a quantum computers. Among those computational problems *Approximate Shortest Independent Vector Problem* ($SIVP_\gamma$) one of the most well studied problems. LWE and SIS are two average-case SIVP problems, that we use in this paper.

Definition 1 (Learning With Errors (LWE)) For integers $n, m \geq 1$, and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and the Gaussian error distribution χ, the distribution $A_{\mathbf{s},\chi}$ is obtained by sampling $\mathbf{a} \in \mathbb{Z}_q^n$ and $e \leftarrow \chi$, and outputting the pair $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e)$. LWE problem (decision-LWE problem) requires to distinguish LWE samples from truly random samples $\leftarrow \mathbb{Z}_q^n \times \mathbb{Z}_q$.

For a prime power q, $b \geq \sqrt{n}\omega(\log n)$, and distribution χ, solving $LWE_{n,q,\chi}$ problem is at least as hard as solving $SIVP_\gamma$, where $\gamma = \tilde{\mathcal{O}}(nq/b)$ [8].

Since the LWE problem was defined [15], it has been extensively studied and used. In this paper, we use the decisional version of the LWE problem.

Definition 2 (Small Integer Solution (SIS)) Given uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find non-zero vector $\mathbf{x} \in \mathbb{Z}^m$, such that $\mathbf{A} \cdot \mathbf{x} = 0 \mod q$ and $\|\mathbf{x}\|_\infty \leq \beta$.

For any m, $\beta = \mathsf{poly}(n)$, and $q > \sqrt{n\beta}$, solving $SIS_{n,m,q,\beta}$ problem with non-negligible probability is at least as hard as solving $SIVP_\gamma$ problem, for some $\gamma = \beta \cdot \tilde{\mathcal{O}}(\sqrt{nm})$ [8], [10].

2.3 Lattice Related Algorithms

We use the below defined preimage sampleable trapdoor functions (PSTFs) in our construction.

Lemma 1 (TrapGen[20]) For a odd integer $q \geq 3$ and $m = \lceil 6n \log q \rceil$ this algorithm outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{T}_\mathbf{A} \in \mathbb{Z}_q^{m \times m}$ for $\Gamma_q^\perp(\mathbf{A})$ such that $||\widetilde{\mathbf{T}_\mathbf{A}}|| \leq O(\sqrt{n \log q})$ and $||S|| \leq O(n \log q)$ with all but negligible probability in n.

Lemma 2 (SamplePre [8]) On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a trapdoor basis $\mathbf{R}$, a target image $\mathbf{u} \in \mathbb{Z}_q^n$,

and the standard deviation $\sigma \geq \omega(\sqrt{\log m})$, the PPT algorithm $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{u}, \sigma)$ outputs a sample $\mathbf{e} \in \mathbb{Z}^m$ from a distribution that is within negligible statistical distance of $D_{\Lambda \frac{\mathbf{u}}{q}(A),\sigma}$.

Lemma 3 (ExtBasis [5]) ExtBasis is a probabilistic-polynomial-time algorithm that takes a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$, whose first m columns span $\mathbb{Z}_q^n$, and a basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, where $\mathbf{A}$ is the left $n \times m$ sub-matrix of $\mathbf{B}$, as inputs, and outputs a basis $\mathbf{T_B}$ of $\Lambda_q^{\perp}(\mathbf{B})$ with $||\widetilde{\mathbf{T_B}}|| \leq ||\widetilde{\mathbf{T_A}}||$.

2.4 Attribute-Based Encryption

Setup: This algorithm takes the security parameter λ as inputs, and generates a public parameter $\mathbf{PK}$ and a master secret key $\mathbf{MK}$.

KeyGen: For a given public parameter $\mathbf{PK}$, a master secret key $\mathbf{MK}$, and an attribute set $\mathcal{S}$ for a user, this algorithm outputs a user private key $\mathbf{SK}$ associated with $\mathcal{S}$.

Encrypt: On input the public parameter $\mathbf{PK}$, and an access tree (policy) $\mathcal{W}$, and a message m, this algorithm outputs a ciphertext C.

Decrypt: On input a user private key $\mathbf{SK}$ and a ciphertext C for a message m, this algorithm outputs the message m, if the user attribute set $\mathcal{S}$ can satisfy the given policy.

3. Group Oriented Attribute Based Encryption (GO-ABE) Scheme of Li et al.

One of the security requirement of traditional ABE is collusion resistance; no group of users able to combine their private keys to decrypt a ciphertext. However, sometimes ABE is not preferable in real life such as it is required users collaboration. In 2015, Li et al. [9] suggested GO-ABE, which enable users to collaborate to decrypt a ciphertext when there is no user who can alone satisfy the access policy. Thus GO-ABE support emergency situations such as accessing patient data in PHR system. In the GO-ABE scheme suggested by Li et al. [9], the users belong to a specific group, and only users from the same group can pool their attributes to satisfy the access tree. However, no user will reveal their private keys.

3.1 GO-ABE

Definition 3 A group-oriented attribute-based encryption scheme is parameterized by a universal set of attributes $\mathbb{A}$, a space of group identities $\mathbb{G} = g_1, g_2, \ldots, g_n$, and a message space $\mathbb{M}$, and has the following algorithms.

Setup: This randomized algorithm takes inputs as only the security parameter, and outputs a public parameter $\mathbf{PK}$ and a master secret key $\mathbf{MK}$.

Encryption: On input, the public parameter $\mathbf{PK}$, a message $M \in \mathbb{M}$, and a set of attributes (access structure) $\mathcal{W}$, this algorithm outputs a ciphertext C for message m.

KeyGen: On input, the public parameter $\mathbf{PK}$, the master secret key $\mathbf{MK}$, a group id g, and an attribute set $\mathcal{S}$, this algorithm outputs a decryption key $\mathbf{SK}_{\mathcal{S}}^{g}$.

Decryption: On input, the ciphertext C, that was encrypted under a set of attributes $\mathcal{W}$, the public parameter $\mathbf{PK}$, and a set of users from the same group g, this algorithm pools the user attribute sets as $U = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots, \mathcal{S}_N$ to generate a decryption key $\mathbf{SK}_U^g$ and outputs the message m if $|\mathcal{W} \cap U| \geq t$, where t is the threshold value.

3.2 Security Definition: Selective-Set Model for GO-ABE

The selective set model game captures the indistinguishability of challenging ciphertext. The adversary's goal is to determine which of the two messages is encrypted.

Int: The adversary declares the attribute set $\mathcal{W}$ that he wishes to be challenged upon.

Setup: The challenger generates a public parameter $\mathbf{PK}$ and a master secret key $\mathbf{MK}$ executing Setup and sends $\mathbf{PK}$ to the adversary.

Phase 1: The adversary queries the private secret keys $\mathbf{SK}_{\mathcal{S}_i}^{g}$ for different attribute sets $\mathcal{S}_i$ with a group id $g \in \mathbb{G}$, where $|\mathcal{S}_i \cap \mathcal{W}| < t$ for all i.

At the end of Phase 1, $|U_i \cap \mathcal{W}| < t$, where $U_i = \mathcal{S}_1 \cup \mathcal{S}_2, \ldots, \mathcal{S}_N$ is the union of attribute sets all from the group g.

Challenge: The adversary sends two messages M_0 and M_1 whose lengths are the same. The challenger selects $b \leftarrow \{0, 1\}$ and encrypts M_b with $\mathcal{W}$. Then he passes the generated ciphertext C to the adversary.

Phase 2: Phase 1 is repeated with the same conditions.

Guess: The adversary outputs a guess b'.

The advantage of the adversary winning the game is $Pr[b' = b] - 1/2$.

Definition 4 The GO-ABE scheme is secure in the Selective-set model of security if all polynomial-time adversaries have at most negligible advantage in the above Selective-set game.

4. Our GO-ABE lattice-based construction

4.1 Description

Let $\mathbb{A}$ be the universal attribute set of size ℓ. Each attribute has a matrix $\mathbf{A}$ which is publicly available. Thus the public parameters $\mathbf{PK}$ consists of ℓ matrices $(\mathbf{A}_1, \ldots, \mathbf{A}_\ell)$ and a vector $\mathbf{u}$. The master secret key $\mathbf{MSK}$ consists of the trapdoors $(\mathbf{T}_1, \ldots, \mathbf{T}_\ell)$ corresponding to each matrix $\mathbf{A}_i$. The trapdoor $\mathbf{T}_i$ is used to derive secret key $\mathbf{x}_i$ using the Gaussian sampling algorithm. In our scheme, we take the vector $g \in \mathbb{G}$ as the group id and each user id $d \in \mathbb{N}$. Each group has a uniformly random matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ with related trapdoor $\mathbf{T}_G$ obtained from $\mathsf{TrapGen}(n, m, q)$, two other randomly selected matrices $\mathbf{G}_0, \mathbf{G}_1 \in \mathbb{Z}_q^{m \times n}$, and a uniformly random vector $\mathbf{g} \in \mathbb{Z}_q^n$. We assume the public key $\mathbf{GPK} = (\mathbf{G}, \mathbf{G}_0, \mathbf{G}_1, \mathbf{g})$ and secret key $\mathbf{T}_G$ for each group with id g is selected and stored such that only the authority with $\mathbf{MSK} = (\mathbf{T}_1, \ldots, \mathbf{T}_\ell)$ can access $\mathbf{T}_G$. Thus at the key generation for a set of attributes, the authority can compute the secret key relating to the group. First using the

SSS scheme authority shares the public key $\mathbf{u}$ into ℓ shares, where ℓ is the size of the universal attribute set. Next based on the group and then based on the user id secret keys are computed for each possessing attribute by a user. Since the secret keys are based on the group only the same group of users can pool to regenerate or satisfy $\mathbf{u}$.

Satisfying the above requirements we employ Shamir's $k-out-of-\ell$ secret sharing scheme. Inspired by the work of Agrawal et al. [3] to answer the issues of correctness and security challenges, fractional Lagrangian coefficients are used in reconstructing the public key $\mathbf{u}$. As a result, we take sufficiently large constant D as in [3] to multiply with the noise vector when generating the ciphertext.

To keep the scheme simple, we take a message bit $M \in \{0,1\}$ as in [3].

4.2 Construction of Algorithms

Let $\lambda \in \mathbb{Z}^+$ be a security parameter. Let $n = n(\lambda)$, $m = m(\lambda)$ be two positive integers and $q = q(\lambda)$ be a prime. Let $\sigma = \sigma(\lambda)$ be a Gaussian parameter.

- $\mathsf{Setup}(1^\lambda)$: On input a security parameter λ the algorithm outputs the public parameters $\mathbf{PK}$ and the master secret key $\mathbf{MSK}$.
  (1) Obtain uniformly random matrices $\mathbf{A}_{i=1}^{\ell} \in \mathbb{Z}_q^{n\times m}$ and corresponding trapdoors $\mathbf{T}_{i=1}^{\ell}$ executing $\mathsf{TrapGen}(n,m,q)$ for all attributes in $\mathbb{A}$.
  (2) Select uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$.
  (3) Output $\mathbf{PK} = (\{\mathbf{A}\}_{i\in[\ell]}, \mathbf{u})$ and $\mathbf{MSK} = \{\mathbf{T}\}_{i\in[\ell]}$.
- $\mathsf{Encrypt}(\mathbf{PK}, m, \mathcal{W})$: This algorithm takes the public parameter $\mathbf{PK}$, a message $M \in \{0,1\}$, and a policy $\mathcal{W}$ with attribute size w, and outputs the ciphertext C as below.
  (1) Let $D \stackrel{\text{def}}{=} (\ell!)^2$.
  (2) Select a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e}_i \in \mathbb{Z}_q^m$ for $i \in [w]$, and $e \in \mathbb{Z}_q$.
  (3) Set $\mathbf{c}_1 = \mathbf{A}_i^T\mathbf{s} + D\mathbf{e}_i$ for $i \in [w]$, $\mathbf{c}_2 = \mathbf{u}^T\mathbf{s} + De + M\lfloor q/2 \rfloor$.
  (4) Output $C = (\mathbf{c}_1, \mathbf{c}_2)$.
- $\mathsf{KeyGen}(\mathbf{PK}, \mathbf{MSK}, g, \mathcal{S})$: On input the public parameter $\mathbf{PK}$, the master key $\mathbf{MSK}$, a group id g which the user belongs to, and a user id d with the possessing attribute set $\mathcal{S}$, this algorithm outputs private key (decryption key) $\mathbf{SK}_{\mathcal{S}}^{g}$ which consists of $\mathbf{sk}_i^{g,d}$ for each attribute $i \in \mathcal{S}$.
  (1) Select the group public $\mathbf{GPK} = (\mathbf{G}, \mathbf{G}_0, \mathbf{G}_1, \mathbf{g})$, and secret key $\mathbf{GSK} = \mathbf{T}_G$ related to the group id g.
  (2) Select a fresh positive integer $d \in \mathbb{N}$ as the user id possessing $\mathcal{S}$.
  (3) Using Shamir secret sharing (SSS) scheme construct ℓ shares of vector $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{Z}_q^n$ (applying SSS scheme for each co-ordinates of $\mathbf{u}$ independently). Be precise, for each $j \in [n]$ select a uniformly random polynomial $p_j \in \mathbb{Z}_q[x]$ of degree $k-1$ such that $p_j(0) = u_j$. Here k is the threshold value. Construct the $j-th$ share vector $\hat{\mathbf{u}}_j = (\hat{u}_{j,1}, \ldots, \hat{u}_{j,n}) = (p_1(j), p_2(j), \ldots, p_n(j)) \in \mathbb{Z}_q^n$.
  Calculate fractional Lagrangian coefficients L_j such that $\mathbf{u} = \sum_{j\in J} L_j \cdot \hat{\mathbf{u}}_j (\mod p)$. Note that for all $J \subset [\ell]$ such that $|J| \geq k$ [3].
  (4) For each attribute $i \in \mathcal{S}$, using $\mathsf{SamplePre}(\mathbf{A}_i, \mathbf{T}_i, \hat{\mathbf{u}}_i - \mathbf{g}, \sigma)$ find $\mathbf{v}_i \in \mathbb{Z}_q^m$ such that $\mathbf{A}_i \cdot \mathbf{v}_i = \hat{\mathbf{u}}_i - \mathbf{g}$.
  (5) For the user with id d compute $\mathbf{G}_d = [\mathbf{G}|\mathbf{G}_0 + d\mathbf{G}_1] \mathbb{Z}_q^{m\times 2n}$ and obtain a short basis $\mathbf{T}_d$ for the lattice $\Lambda^{\perp}(\mathbf{G}_d)$ executing $\mathsf{ExtBasis}(\mathbf{T}_G, \mathbf{G}_d)$.
  (6) Then for each attribute $i \in \mathcal{S}$ obtain $\mathbf{x}_i^d \leftarrow \mathsf{SamplePre}(\mathbf{G}_d, \mathbf{T}_d, \mathbf{v}_i, \sigma)$, such that $\mathbf{G}_d \cdot \mathbf{x}_i^d = \mathbf{v}_i$. Note that, $(\mathbf{A}_i \cdot (\mathbf{G}_d \cdot \mathbf{x}_i^d)) + \mathbf{g} = \hat{\mathbf{u}}_i$.
  (7) Output $\mathbf{SK}_{\mathcal{S}}^{g} = ((\mathbf{x}_1^d, \ldots, \mathbf{x}_s^d), d)$, where $s = |\mathcal{S}|$.
- $\mathsf{Decrypt}(\mathbf{PK}, C, U^g)$: On input the public parameter $\mathbf{PK}$, the ciphertext C, and the set of users U^g from the same group $\mathbb{G}$ with group id g, this algorithm executes as below and returns a message m if the attributes satisfy the decryption policy. That is , $|\mathcal{W} \cap U| \geq k$ and $U = S_1 \cup S_2 \cup, \ldots, \cup, S_N$. Note that the secret keys of users $\mathbf{SK}_{\mathcal{S}_i}^{g}$ are only known to the owners.
  (1) Select an arbitrary subset $\mathcal{S}$ with size k of $\mathcal{W} \cap U$.
  (2) Each user computes $\mathbf{G}_d$ using his id d and publishes $\mathbf{y}_i = (\mathbf{G}_d \cdot \mathbf{x}_i)$ for $i \in [k]$
  (3) The ciphertext can be decrypted as follows.
    – Calculate the fraction Lagrangian coefficients L_i;
    $\sum_{i\in[k]} L_i \mathbf{A}_i \mathbf{y}_i = \mathbf{u} \mod q$.
    – Compute $r \leftarrow \mathbf{c}_2 - ((k\times\mathbf{g})^T + \sum_{i\in[k]} L_i \mathbf{y}_i^T \mathbf{c}_1) (\mod q)$, where $\mathbf{g}$ is the unique key (part of the group public key) of the group with id g.View it as $r \in [-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor) \subset \mathbb{Z}$.
    – If $|r| < q/4$, output 0, else output 1 as message M.

5. Analysis of the Scheme

In this section, we prove the correctness and security of the lattice-based construction of GO-ABE scheme. GO-ABE scheme presented by Li et al. [9] requires the users to be from the same group and not to reveal their secret keys. Thus we shared the public key $\mathbf{u}$ to ℓ shares, where ℓ is the universal attribute size and computed the secret key for each user possessing attribute based on the relevant share of $\mathbf{u}$ and the group keys. Thus unless all the users are from the same group they cannot collude together. Moreover, we enabled the users only to pool the computed output of their secret keys, not the secret keys. Thus it ensures that the secret keys are secured.

In the below we discuss the correctness and security proofs of the construction.

5.1 Correctness

For the proof of correctness of the decryption, we only

need to consider the case $|J| \geq k$. Let L_j be the fractional Lagrangian coefficients as discussed before.

$r \leftarrow \mathbf{c}_2 - ((k \times \mathbf{g})^T + \sum_{i\in[k]} L_i \mathbf{y}_i^T \mathbf{c}_1)(\mod q)$.

Here,

$\mathbf{c}_2 = \mathbf{u}^T\mathbf{s} + \mathbf{e} + b\lfloor q/2 \rfloor$

k is the threshold value.

$\mathbf{g}$ is the group-related public key vector.

$\mathbf{y}_i = (\mathbf{G}_d \cdot \mathbf{x}_i)$

$\mathbf{c}_1 = \mathbf{A}_i^T\mathbf{s} + \mathbf{e}_i$

Thus we can write,

$r \leftarrow \mathbf{c}_2 - ((k \times \mathbf{g})^T + \sum_{i\in[k]} L_i \mathbf{y}_i^T \mathbf{c}_1)(\mod q)$ as

$r \leftarrow (\mathbf{u}^T\mathbf{s} + De + b\lfloor q/2 \rfloor) - ((k \times \mathbf{g})^T + (\sum_{i\in[k]} L_i(\mathbf{G}_d \cdot \mathbf{x}_i)^T \mathbf{A}_i^T\mathbf{s} + D\mathbf{e}_i))$.

In simple,

$r \leftarrow b\lfloor q/2 \rfloor + (\mathbf{u}^T\mathbf{s} - ((k \times \mathbf{g})^T + (\sum_{i\in[k]} L_i(\mathbf{G}_d \cdot \mathbf{x}_i\mathbf{A})_i^T\mathbf{s}))) + (Dx - \sum_{i\in[k]} DL_i\mathbf{x}_i^T\mathbf{e}_i)(\mod q) \approx b\lfloor q/2 \rfloor$.

Here, $(Dx - \sum_{i\in[k]} DL_i\mathbf{x}_i^T\mathbf{e}_i) \approx 0$.

This proves the correctness of our scheme construction.

5.2 Security Proof

We show the lattice-based construction of GO-ABE provides ciphertext privacy in the Selective-Set model under the hardness of the LWE problem.

Theorem 1 If there is an adversary $\mathcal{A}$ with advantage $\epsilon > 0$ against the selective-set model for the GO-ABE scheme, then there exists a PPT algorithm $\mathcal{B}$ that can solve the decision-LWE problem.

Proof. The simulator $\mathcal{B}$ uses the adversary $\mathcal{A}$ to distinguish LWE oracle $\mathcal{O}$. First $\mathcal{B}$ queries the LWE oracle $\mathcal{O}$ for $(\ell m+1)$ times and obtain LWE samples $(\mathbf{a}_k, b_k) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $k \in \{0, 1, 2, \ldots, m\}$. Then $\mathcal{B}$ proceeds as below.

- **Init**: $\mathcal{A}$ announces the challenging access structure $\mathcal{W}^*$ to $\mathcal{B}$.
- **Setup**: $\mathcal{B}$ prepares the public keys as follows.
  (1) Choose ℓ matrices $\hat{\mathbf{A}}_i, i \in [\ell]$ from LWE challenge $\{(\mathbf{a}_0, b_0), (\mathbf{a}_i^1, b_i^1), (\mathbf{a}_i^2, b_i^2), \ldots, (\mathbf{a}_i^m, b_i^m)\}_{i\in[\ell]}$.
  (2) Select ℓ matrices $\mathbf{A}_i$ and trapdoors $\mathbf{T}_i$ using $\mathsf{TrapGen}$.
  (3) Set vector $\mathbf{u}$ from LWE challenge $\mathbf{a}_0$.
  (4) Give public parameters to $\mathcal{A}$
- **Phase 1**: $\mathcal{B}$ answers each private key query for attribute set $\mathcal{S}$ as follows.
  (1) Let $\mathcal{S} \cup \mathcal{W}^* := I \subset [\ell]$ and let $|I| = t < k$.
  (2) Represents the shares of $\mathbf{u}$ as $\hat{\mathbf{u}}_i = \mathbf{u} + \mathbf{v}_1 i + \mathbf{v}_2 i^2 + \ldots + \mathbf{v}_{k-1} i^{k-1}$ where $\mathbf{v}_1, \ldots \mathbf{v}_{k-1}$ are vectors of length n each.
  (3) For all $i \in [t]$ select $\mathbf{x}_i$ and set $\hat{\mathbf{u}}_i := \hat{\mathbf{A}}_i\mathbf{x}_i$.
  (4) For all $j \in [k-1-t]$ invoke $\mathsf{SamplePre}(\mathbf{A}_i, \mathbf{T}_j, \hat{\mathbf{u}}_j, \sigma)$
  (5) Return $(\mathbf{x}_1, \ldots, \mathbf{x}_\ell)$.
- **Challenge**: $\mathcal{A}$ outputs challenge messages M_0 and M_1. The simulator $\mathcal{B}$ responds with a challenge ciphertext for $\mathcal{W}^*$ as follows.
  (1) Let $c_1 = (Db_i^1, Db_i^2, \ldots, Db_i^m)$ for $i \in [\ell]$.
  (2) Let $c_2 = D\mathbf{a}_0 + M_b\lfloor q/2 \rfloor$.
- **Phase 2**: The simulator repeats Phase 1 under the same conditions.
- **Guess**: The adversary $\mathcal{A}$ outputs a guess b'. If $b = b'$ then $\mathcal{A}$ wins the game.

6. Conclusion

In this paper, we provide a construction of the GO-ABE scheme from lattices that supports users from the same group to pool their attributes anonymously (without revealing their secret keys) to satisfy a given access tree. Since we used lattice cryptography, our scheme is quantum resistant. However, some limitations need to discuss in the future. For instance, the users can pool their attributes even not in an emergency in the GO-ABE scheme. We believe there should be control based on the situation.

References

[1] Attrapadung, N., Libert, B., de Panafieu E.: Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In: PKC 2011. Lecture Notes in Computer Science, vol 6571, pp 90-108, (2011).

[2] Agrawal, S., Boneh, D., Boyen X.: Efficient Lattice (H)IBE in the Standard Model. In: EUROCRYPT 2010 Lecture Notes in Computer Science, vol 6110, pp 553-572, (2010).

[3] Agrawal, S., Boyen, X., Vaikuntanathan, V., Voulgaris, P., Wee, H.: Functional encryption for threshold functions (or fuzzy IBE) from lattices. In: PKC 2012 Lecture Notes in Computer Science, vol 7293, pp 280-297 (2012).

[4] Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE symposium on security and privacy (SP'07). IEEE, pp. 321-334, (2007).

[5] Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: EUROCRYPT 2010. Lecture Notes in Computer Science, vol 6110, pp 523-552, (2010).

[6] Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: Proceedings of the 14th ACM conference on Computer and communications security, pp. 456-465, (2007).

[7] Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M.: A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length. In: ISPEC 2009. Lecture Notes in Computer Science, vol 5451, pp 13-23, (2009).

[8] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the fortieth annual ACM symposium on Theory of computing (STOC '08), (2008)

[9] Li, M., Huang, X., Liu, J.K., Xu, L.: GO-ABE: Group-Oriented Attribute-Based Encryption. In: NSS 2015. Lecture Notes in Computer Science, vol 8792, pp 260-270, (2014).

[10] Ling, S., Nguyen, K., Wang, H.: Group Signatures from Lattices: Simpler, Tighter, Shorter, Ring-Based. In: PKC 2015. Lecture Notes in Computer Science, vol 9020, pp 427-449, (2015)

[11] Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: EUROCRYPT 2010. Lecture Notes in Computer Science, vol 6110, pp 62-91, (2010).

[12] Lewko, A., Waters, B.: Decentralizing Attribute-Based Encryption. In: EUROCRYPT 2011. Lecture Notes in Computer Science, vol 6632, pp 568-588, (2011).

[13] Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption. In: IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, pp 131-143, (2013).

[14] Li, Q., Xiong, H., Zhang, F., Zeng, S.: An expressive decentralizing kp-abe scheme with constant-size ciphertext. In: IJ Network Security, 15(3), pp 161-170 (2013).

[15] Regev, Oded.: New lattice-based cryptographic constructions. In: Journal of the ACM (JACM), vol 51 number 6, pp 899-942, (2004)

[16] Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In:

EUROCRYPT 2005. Lecture Notes in Computer Science, vol 3494, pp 457-473, (2005).

[17] Wang, Y., Chen, K., Long, Y., Liu, Z. (2012).: Accountable authority key policy attribute-based encryption. In: Science China Information Sciences, 55(7), pp 1631-1638, (2012).

[18] Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: EUROCRYPT 2005. EUROCRYPT 2005. Lecture Notes in Computer Science, vol 3494, pp 114-127(2005).

[19] Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: PKC 2011. Lecture Notes in Computer Science, vol 6571, pp 53-70, (2011).

[20] Wang, Y.: Lattice Ciphertext Policy Attribute-based Encryption in the Standard Model. Int. J. Netw. Secur., 16, pp 444-451, (2014).

[21] Yu, S., Wang, C., Ren, K., Lou, W.: Achieving secure, scalable, and fine-grained data access control in cloud computing. In 2010 Proceedings IEEE INFOCOM , pp 1-9, (2010).