Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Efficient lattice-based revocable attribute-based encryption against decryption key exposure for cloud file sharing

Boxue Huang[1], Juntao Gao[1*] and Xuelian Li[2]

Abstract

Cloud file sharing (CFS) has become one of the important tools for enterprises to reduce technology operating costs and improve their competitiveness. Due to the untrustworthy cloud service provider, access control and security issues for sensitive data have been key problems to be addressed. Current solutions to these issues are largely related to the traditional public key cryptography, access control encryption or attribute-based encryption based on the bilinear mapping. The rapid technological advances in quantum algorithms and quantum computers make us consider the transition from the tradtional cryptographic primitives to the post-quantum counterparts. In response to these problems, we propose a lattice-based Ciphertext-Policy Attribute-Based Encryption(CP-ABE) scheme, which is designed based on the ring learing with error problem, so it is more efficient than that designed based on the learing with error problem. In our scheme, the indirect revocation and binary tree-based data structure are introduced to achieve efficient user revocation and dynamic management of user groups. At the same time, in order to further improve the efficiency of the scheme and realize file sharing across enterprises, the scheme also allows multiple authorities to jointly set up system parameters and manage distribute keys. Furthermore, by re-randomizing the user's private key and update key, we achieve decryption key exposure resistance(DKER) in the scheme. We provide a formal security model and a series of security experiments, which show that our scheme is secure under chosen-plaintext attacks. Experimental simulations and evaluation analyses demonstrate the high efficiency and practicality of our scheme.

Keywords Cloud file sharing, Attribute-based encryption, Dynamic management, Multi-authority, Decryption key exposure

Introduction

Cloud file sharing (CFS) has been widely used in current cloud services. According to Gallup's report, 81% of employees of 60 million full-time employees are choosing to work from home remotely work or mixed work (part-time working from home) by reason of the COVID-19 epidemic. The CFS has the advantages of flexible use and low cost. Employees can access cloud data from any location through any internet device (e.g. mobile phone, tablet, laptop, etc.) to meet flexible remote work needs. At the same time, cloud storage servers can meet the high storage requirements of users and enterprises, and provide low-cost and diversified cloud services. It greatly saves the company's data storage cost and improves the competitiveness of the enterprise. As a result, CFS services (such as Amazon WorkDocs, iCloud, Dropbox, Google Drive, OneDrive, Mega, etc.) have become the first choice for increasingly competitive individuals and businesses.

As a leading cloud storage platform, MEGA has more than 250 million users, and aims to provide users with

*Correspondence:
Juntao Gao
jtgao@mail.xidian.edu.cn
[1] School of Telecommunications Engineering, Xidian University, Xi'an, China
[2] School of Mathematics and Statistics, Xidian University, Xi'an, China

end-to-end encryption and information security assurance controlled by users. However, in a recent study [1], Backendal et al. found a significant shortcomings in the platform's cryptographic architecture. They have carried out five attacks against the RSA encryption algorithm in MEGA, such as RSA key recovery attack, framing attack, integrity attack, etc., to destroy the user data integrity to some extent. Through the RSA key recovery attack, the attacker can recover the RSA private key after 1023 client login attempts, while using quantum cryptanalysis can reduce it to 512 attempts, so that the private key can be recovered more quickly.

Hence, how to protect the user's data in the cloud from being tampered with, stolen, or illegally accessed by other users has always been a hot research issue. In order to provide fine-grained access control, Attribute-Based Encryption(ABE) has been widely used in various CFS systems [2–6]. In the ciphertext-policy ABE(CP-ABE), the access policy is associated with the ciphertext, and the attribute is associated with the key, which makes CP-ABE more focused on the role-based access control than the key-policy ABE(KP-ABE), Hence CP-ABE is more suitable for CFS systems.

Currently, there are several security and access control issues on the CP-ABE scheme proposed for the CFS system.

Resistant to quantum attacks : With the further study of quantum computers and quantum attack algorithms, the security of traditional ABE based on bilinear mapping has been seriously challenged. In [7], the author also pointed out that cryptographic algorithms and cryptography-related devices should immediately start to transition to the post-quantum cryptography suite of algorithms. Otherwise, some sensitive documents, such as business secrets, medical records, national security documents and other documents that have a long shelf life can be leaked out since the transition process could take multi-decade, by which time the quantum computer may have been mastered by the adversary. Currently, cryptographic algorithms based on the Learning With Error (LWE) problem in lattice are generally considered to be effective against quantum attacks. However, the large parameter size and low efficiency of LWE-based ABE schemes make the cryptographic researcher consider alternative ABE schemes designed based on the Ring Learning With Error (RLWE) problem with smaller parameter size and higer efficiency. Current RLWE-based ABE schemes are deficient in dynamic management of user groups, distributed management and the decryption key exposure resistance. Therefore, the design and security analysis of RLWE-based CP-ABE schemes with multiple properties for CFS system become a challenging issue.

Dynamic management of user groups : In practical applications, the CFS system usually changes users' data access policy when users' position or role change, such as revoking the user's access privileges to specific files and private data. Users' privileges revocation can be divided into the direct revocation and indirect revocation. In the direct revocation, the data owner can directly revoke the user's permissions. However, the direct revocation is not applicable in CFS because the data owner must always be online and maintain an up-to-date revocation list at all times. Indirect revocation is suitable for CFS system, in which the attribute authority can periodically broadcasts the key-updating material to users in the CFS system, and users who have not been revoked can update their credentials. However, indirect revocation also faces a problem, that is, the attribute authority needs to generate key update materials for each user in the revocation process. Although Yang et al. [8] and Wang et al. [9] used the binary tree structure to reduce the size of the key update material for each user, this is still a heavy burden for the authorization authority in the CFS system with a large number of users. Therefore, how to implement user revocation securely and efficiently is still an urgent problem.

Decryption key exposure attacks : In the ABE scheme proposed for CFS system, the user's decryption key is often used for the decryption of private data. Once the decryption key is exposed, the private data will be opened for the adversary. Hence the key exposure attack is a main secure threat to the CFS system. In the indirect revocation cryptosystems, decryption key exposure attack is a common attack method. The attacker can obtain the user's long-term private key by calculating the old and leaked decryption key and the update key transmitted from the public channel. This means that the attacker can derive all subsequent decryption keys. In ABE schemes based on the bilinear mapping [10, 11], the author proposed the method of key re-randomization to achieve decryption key exposure resistance. In contrast, the lattice-based CP-ABE scheme has a more complex algebraic structure, and it is more difficult to re-randomize its key. Therefore, how to re-randomize the key of RLWE-based CP-ABE to resist decryption key exposure attack remains challenged.

In addition, the lattice-based multi-authority attribute-based encryption(MA-ABE) scheme could be more suitable for a large-scale CFS system than an ABE scheme with only one authority, because a shared file often has access policies that span multiple trust domains. For example, multiple companies may publish attributes as part of a joint project. If a single authority is employed, one company must be required to cede control to another, which has great limitations in practical applications. Current MA-ABE schemes [12, 13] are designed based on the LWE problems. What we concerns is how to design the RLWE-based MA-ABE schemes with the above desired properties.

Related work

Some CFS systems with access control encryption or CP-ABE have been proposed. Zhu et al. [14] proposed an efficient temporary access control encryption scheme for cloud services based on a proxy encryption mechanism and cryptographic integer comparison, while extending the power of attribute expression using a dual comparative expression of integer ranges. Zhu et al. [15] proposed an ABE scheme without pairings for CFS systems, and proved the chosen plaintext security in selective ID model. Zhu et al. [16] proposed a new fuzzy authorization scheme based on CP-ABE scheme and OAuth for cloud data access, and realized automatic revocation by updating the TimeSlot attribute when data owner modifies the data. Finally, the security of the scheme is proven under the d-BDHE assumption. Wang et al. [6] proposed an efficient cloud computing encryption scheme based on file-level attributes. Layered files are encrypted with an integrated access structure. However, the above ABE schemes are all based on bilinear pairing and do not have the ability to resist quantum attacks.

Considering the hardness of lattice problems in post-quantum cryptography, researchers begin to construct lattice-based ABE schemes. Zhang et al. [17] first constructs a CP-ABE scheme based on the LWE assumption. However, due to the introduction of default properties, the parameter size of the overall scheme increases dramatically and the efficiency is low. In order to solve this problem, chen et al. [2] uses a small policy matrix to reduce the cumulative error and improve the efficiency of the scheme. At the same time, they also proposed a new resource sharing framework combined with the ABE scheme. Chen et al. [18] improved on the basis of [17], and implemented a large universe CP-ABE scheme using the full-rank differences function, which can obtain greater efficiency. To further improve efficiency, Gür et al. [19] constructs and implements the CP-ABE scheme from lattices under the RLWE assumption. They employ Gaussian sampling for bigger bases of the gadget matrix, which reduces execution times and storage requirements.

In order to solve the user dynamic management problem in the CFS system, Ibraimi et al. [20] proposed a revocable CP-ABE scheme, but this scheme requires a trusted cloud service provider and requires it to be online for a long time. This does not meet actual needs. Sahai et al. [21] proposed a CP-ABE scheme with revocable storage, which allows a cloud Server to update the ciphertext, but this scheme adds many additional time attributes, resulting in bigger public key and ciphertext. Li et al. [3] proposed an efficient CP-ABE scheme with policy update and file update functions based the CFS system, which effectively reduces the communication and storage cost of the client.

In order to prevent decryption key exposure attacks in CFS systems, Xu et al. [10] proposes a CP-ABE scheme with decryption key exposure resistance. Xu et al. [22] proposed a CP-ABE scheme with decryption key exposure resistance based on cloud storage system, which can effectively deal with both secret key revocation for corrupted users and accidental decryption key exposure for honest users. However, the above revocable CP-ABEs are all designed based on bilinear pairing, and they cannot effectively resist quantum attacks. Wang et al. [9] and yang et al. [23] proposed two LWE-based revocable CP-ABE schemes. However, the public key size of scheme [9] is large, and the revocation workload of scheme [23] increases linearly with the increase of the number of system users, which makes the efficiency and scalability of the two schemes low. Takayasu et al. [24] and Dong et al. [25] proposed lattice-based IBE scheme and KP-ABE scheme for decryption key exposure resistance, respectively. But in the lattice-based CP-ABE scheme, which is more suitable for CFS system, there is no formal solution.

In order to further improve the efficiency of the CFS system, reduce the workload of a single AA on the enterprise side, and realize access policies across multiple trust domains at the same time, Chase et al. [26] proposed the multi-attribute authority ABE scheme for the first time. The scheme includes a central authority and multiple attribute authorities, and the attributes in the system are jointly managed by multiple attribute authorities. Subsequently, Rouselakis et al. [27] proposed an MA-ABE scheme based on prime order bilinear groups to further improve the efficiency. To deal with quantum attacks, Zhang et al. [13] proposed lattice-based MA-ABE for the first time. However, since this scheme is based on the assumption of learning with errors, the efficiency of this scheme is low.

Technical challenge

Current lattice-based CP-ABE schemes for CFS system, especialy for LWE-based CP-ABE scheme, suffer from large parameter size, low efficiency or various cryptographic attacks, such as decryption key exposure attack, collusion attack etc.. When we design an RLWE-based CP-ABE with multiple properties, such as, multiple authorities, decryption key exposure resistance, collusion resistance, user group's dynamic management, attribute revocation etc., several technical chanllenges loom over us. Specifically, when designing an RLWE-based CP-ABE with multiple authorities and decryption key exposure resistance, we have to consider a new type of adversary who masters more information on the private keys of some attributes. The adversary could corrupt some AAs to get some information on the private keys of some attributes. Simultaneously, the adversary can get some addtional infomation leaked out from the uncorrupted

AAs due to the decryption key exposure. As the adversary gains more information, the security model becomes more complex. Hence the security proof in our scheme is more difficult than that in the scheme with one single authority, or the scheme with only decryption key exposure resistance. Furthermore, in the traditional ABE schemes based on bilinear pairing, the resistance to decryption key exposure attack is simply implemented by introducing a random number in the decryption key update. However, the lattice-based CP-ABE scheme is usually designed based on a more complex algebraic structure, which makes the original method(by only introducing a random number in the re-randomization process) for decryption key update invalid. Therefore, in order to get the new RLWE-based MA-ABE suitable for the secure CFS system, we need to define new security model, specify the limits of the new adversary and develop new technique to resist the decryption key exposure attack.

Our contribution

In this paper, we designed an RLWE-based Revocable and Multi-authority CP-ABE scheme (RM-CP-ABE) with decryption key exposure resistance for quantum secure CFS system, whose features includes protection of sensitive data from privacy leakage and quantum attacks, dynamic management of user groups, distributed architecture, decryption key exposure resistance, etc. Our work is summarized as follows:

- Many existing CFS systems only provide role-based access control and are not resistant to quantum attacks. We design a CP-ABE scheme based on RLWE to achieve more fine-grained access control and CFS privacy data security under quantum attacks. In order to resist the decryption key attack, we propose a new method applicable to lattice algebraic structure for re-randomizing the key to ensure that the information of the user's private key will not be leaked after the decryption key is exposed. To implement user revocation, we apply a binary tree-based data structure to reduce the cost of key update from linear to logarithmic.
- In order to further improve the system efficiency and realize cross-enterprise file sharing, we implement a multi-authority CP-ABE scheme with a distributed architecture by applying the shamir threshold secret sharing technique. At the same time, in order to ensure the security of private data under collusion attacks by malicious users and corrupt authorities, we provide a specific security model and prove that our scheme is selectively secure under the assumption of error learning.
- Finally, we implement our scheme and evaluate its key generation, user revocation, encryption, and decryption algorithm time overhead and storage overhead in comparison with related schemes. The simulation show that our scheme has high efficiency.

Preliminaries

Notations

In this paper, we denote a cyclotomic ring $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1\rangle$, where each element is a polynomial with integer coefficients and degree at most $n-1$. For an integer $q \geq 2$, we let $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ be a ring in which arithmetic operations on polynomial coefficients are performed modulo q, and the coefficients are integers in the interval $(\lfloor -q/2 \rfloor, \lfloor q/2 \rfloor)$. $\mathcal{R}_q^{1\times m}$, $\mathcal{R}_q^m$, and $\mathcal{R}_q^{m\times m}$ represent row vectors, column vectors, and matrices consisting of elements in $\mathcal{R}_q$. We define $Tran_{V\to M}$ as a function that maps from vector $\mathbf{a} \in \mathcal{R}_q^m$ to matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$. In detail, we expand the coefficients of each polynomial element in vector $\mathbf{a}$ to an n-dimensional row vector in $\mathbb{Z}_q$. Similarly, we define $Tran_{M\to V}$ as a function that maps from matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ to vector $\mathbf{a} \in \mathcal{R}_q^m$, which can be regarded as the inverse process of $Tran_{V\to M}$. In this paper, we utilize a full-rank difference (FRD) function H that maps a random element in $\mathcal{R}_q$ associated with an attribute or time to a matrix in $\mathbf{Z}_q^{n\times n}$.

Lemma 1 [28] *For positive integers q, n with $q > 2$, the FRD function has the following properties*:

- For any $a, b \in \mathcal{R}_q$ and $a \neq b$, the matrix $(\mathrm{H(a)} - \mathrm{H(b)})$ is full rank.
- H is computable in polynomial time $n \log q$.

Lemma 2 (*Leftover Hash Lemma*) *Let q be a positive integer, $m = O(\log_2 q)$ and let $\mathbf{S} \leftarrow \{-1, 1\}^{m\times m}$, $\{\mathbf{A}, \mathbf{B}\} \leftarrow \mathcal{R}_q^{1\times m}$. Then for all $e \in \mathcal{R}_q^m$, we have $(\mathbf{A}, \mathbf{AS}, \mathbf{e}^\top\mathbf{S})$ is statistically close to $(\mathbf{A}, \mathbf{B}, \mathbf{e}^\top\mathbf{S})$.*

Lattice

A full rank lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$. Given a positive intger n and a basis $\mathbf{B} = \{b_1, ..., b_n\} \subseteq \mathbb{R}^n$, the lattice Λ can be represented as

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{Bx} = \sum_{i=1}^{n} x_i \mathbf{b}_i \mid \mathbf{x} \in \mathbb{Z}^n \right\}.$$

For an uniformly randomly matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, the q-ary lattice $\Lambda_q^\perp(\mathbf{A})$ and its coset $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is denoted by

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^{\mathbf{m}} \mid \mathbf{Ax} = 0 \mod q\},$$

$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^{\mathbf{m}} \mid \mathbf{Ax} = \mathbf{u} \mod q\right\}.$$

Discrete Gaussian Distribution: Given a lattice Λ with a parameter $\sigma \in \mathbb{R}$ and a center vector $\boldsymbol{c} \in \mathbb{R}^n$, we denote the n-th dimensional discrete Gaussian distribution by

$$D_{\Lambda,\boldsymbol{c},\sigma} = \frac{\rho_{\boldsymbol{c},\sigma}(\boldsymbol{x})}{\sum_{\boldsymbol{z}\in\Lambda}\rho_{\boldsymbol{c},\sigma}(\boldsymbol{z})}$$

where $\boldsymbol{x} \in \Lambda$ and $\rho_{\boldsymbol{c},\sigma} = \exp\left(-\pi\|\boldsymbol{x}-\boldsymbol{c}\|^2/2\sigma^2\right)$.

Definition 1 (**Ring Learning With Errors**) The Ring Learning With Errors (RLWE) hardness assumption holds if for any PPT adversary $\mathcal{A}$ we have

$$|\Pr[\mathcal{A}(a_i, a_i s + e_i) - \mathcal{A}(a_i, u_i)]| < negl(\lambda)$$

where $\{s, a_i\} \leftarrow \mathcal{R}_q$, $e_i \leftarrow D_{\mathcal{R},\sigma}$, $i = 0, ..., n$ and the probability is taken over the choice of the random coins by the PPT adversary $\mathcal{A}$.

Lattice trapdoors

Lattices with trapdoors are described in [8, 19, 29], and they are statistically indistinguishable from randomly chosen lattices. But trapdoors have some extra information that can be used for efficient solution to the hard problem on lattices. In this paper, we utilize the trapdoor sampler from [19, 29]. The trapdoor sampler contains TrapGen algorithm, SamplePre algorithm and two expansion algorithms: SampleLeft algorithm and SampleRight algorithm, which are given as follows.

- **TrapGen**$(\lambda) \rightarrow (\mathbf{A}, \mathbf{T_A})$: The lattice generation algorithm takes the security parameters λ and σ, modulus q, and a polynomial degree n as input. Then it outputs a vector $\mathbf{A} \in \mathcal{R}_q^{1\times m}$ together with a trapdoor $\mathbf{T_A}$.
- **SamplePre**$(\mathbf{A}, \mathbf{T_A}, u, \sigma, \sigma_s) \rightarrow \mathbf{s}$: The presampling algorithm takes a vector $\mathbf{A} \in \mathcal{R}_q^{1\times m}$, a trapdoor $\mathbf{T_A}$, a target $u \in \mathcal{R}$, and a pair of parameters (σ, σ_s) as input. It outputs a vector $\mathbf{s} \leftarrow D_{\mathcal{R}_q^m}$ such that $\mathbf{A}\cdot\mathbf{s} = u$.
- **SampleRight**$(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, u) \rightarrow \mathbf{s}$: The algorithm takes as input a pair of vectors $(\mathbf{A}, \mathbf{B}) \in \mathcal{R}_q^{1\times m} \times \mathcal{R}_q^{1\times m}$, a trapdoor $\mathbf{T_A}$ and target $u \in \mathcal{R}$, and outputs a vector $\mathbf{s} \leftarrow D_{\mathcal{R}_q^{2m}}$ such that $(\mathbf{A} \mid \mathbf{B})\cdot\mathbf{s} = u$.
- **SampleLeft**$(\mathbf{A}, \mathbf{B}, \mathbf{T_B}, \mathbf{S}, u) \rightarrow \mathbf{s}$: The algorithm takes as input a pair of vectors $(\mathbf{A}, \mathbf{B}) \in \mathcal{R}_q^{1\times m} \times \mathcal{R}_q^{1\times m}$, a trapdoor $\mathbf{T_B}$, a matrix $\mathbf{S} \leftarrow (\pm 1)^{m\times m}$ and target $u \in \mathcal{R}$, and outputs a vector $\mathbf{s} \leftarrow D_{\mathcal{R}_q^{2m}}$ such that $(\mathbf{A} \mid \mathbf{AS} + \mathbf{B})\cdot\mathbf{s} = u$.

Definition 2 (**Well − Sampledness of Vector**)[12] The lattice generation algorithm is said to satisfy the well-sampledness of vector property if for any $\lambda \in \mathbb{N}$, there exists a negligible function $negl(\lambda)$ such that the vector $\mathbf{A}$ is $negl(\lambda)$-close to uniform, where $\mathbf{A}$ is generated by **TrapGen**(λ).

Definition 3 (**Well − Sampledness of Preimage**) [12] The presampling algorithms and two expansion algorithms are said to satisfy the well-sampledness of preimage property if for any $\lambda \in \mathbb{N}$, there exists a negligible function $negl(\lambda)$ such that the distribution of the vector $\mathbf{s_1}$ and $\mathbf{s_2}$ are $negl(\lambda)$-close to $D_{\mathcal{R}_q^m}$ and $D_{\mathcal{R}_q^{2m}}$ respectively, where $\mathbf{s_1}$ is sampled by **SamplePre**$(\mathbf{A}, \mathbf{T_A}, u, \sigma, \sigma_s)$ algorithm and $\mathbf{s_2}$ is sampled by one of the two expansion algorithms with the same parameters.

The binary-tree data structure

The binary tree-based data structure is mainly used to reduce the calculation cost of generating key-updating material from linear to logarithmic. In the binary tree structure, each user is associated with the leaf node. Except for leaf nodes, each node has two children nodes. Let $Path(v)$ be the node set on the path from the root node of the tree to node v, and (v_l, v_r) be the left child node and the right child node of node v respectively if v is a non-leaf node. When a user v_i is revoked at time t_i, add the user to the revocation list rl, and then the subset covering algorithm KUNodes(st, rl, t) [30] is run to fatch the minimum set of key-updating materials related to non-revoked users. The KUNodes(st, rl, t) algorithm is shown as Algorithm 1.

Input:
The binary tree state, st;
The non-empty revocation list, rl;
The time, t;
Output:
the minimum set Y of key-updating materials related to non-revoked users;

```
X, Y ← ∅
for (v_i, t_i) ∈ rl do
    if t_i < t then
        X ← X ∪ Path(v_i)
    end if
end for
for x ∈ X do
    if x_l ∉ X then
        Y ← Y ∪ x_l
    end if
    if x_r ∉ X then
        Y ← Y ∪ x_l
    end if
end for
if Y = ∅ then
    Y ← root
end if
return Y.
```

Algorithm 1 KUNodes(st, rl, t)

System framework

In this section, we present the system model and threat model of CFS with our new RM-CP-ABE scheme. We then provide a formal security model to simulate the adversary's attacks in the model.

RM-CP-ABE scheme

Definition 4 There are eight algorithms in our RM-CP-ABE scheme: GlobalSetup, AuthSetup, KeyGen, KeyUpdate, DKGen, Enc, Dec, and Rev, with associated attribute space $\mathcal{U}$, user space $\mathcal{I}$, time space $\mathcal{T}$, and message space $\mathcal{M}$. The algorithms are defined as follows:

GlobalSetup$(\lambda, N) \rightarrow pp$: The global setup algorithm takes as input the security parameter λ and the number of authorities N, and outputs the public parameters pp.

AuthSetup$(pp, \theta) \rightarrow (pk_\theta, msk_\theta)$: The authority setup algorithm takes as input the public parameters pp, and a number θ to represent the θth authority AA_θ, and outputs a pair of (pk_θ, msk_θ) as the public key and master secret key.

KeyGen$(pk_\theta, msk_\theta, st, S) \rightarrow sk_\theta$: The key generation algorithm takes as input the public key pk_θ, master secret key msk_θ, a state st and a set of attributes S, and outputs the secret key sk_θ associated with AA_θ.

KeyUpdate$(pk_\theta, msk_\theta, rl, st, t) \rightarrow ku_t$: The key update algorithm inputs the public key pk_θ, the master secret key msk_θ, a revocation list rl, and a state st, and a time t and outputs an update key ku_t.

DKGen$(\{sk_\theta, ku_\theta\}_{\theta \in N}) \rightarrow dk$: The decryption key generation algorithm takes the secret key $\{sk_\theta\}_{\theta \in N}$ and the key update $\{ku_\theta\}_{\theta \in N}$ as input. If the user is not revoked, the algorithm outputs the decryption key. Otherwise, it outputs $\perp$.

Enc$(pp, \{pk_\theta\}_{\theta \in N}, \mu, \mathbb{W}) \rightarrow CT$: The encryption algorithm takes as input the public parameters pp, the public key $\{pk_\theta\}_{\theta \in N}$, a message μ, and an access policy $\mathbb{W}$ and outputs a ciphertext CT.

Dec$(CT, dk) \rightarrow \mu$: The decryption algorithm takes as input the ciphertext CT and the decryption key dk. If the attribute set S corresponding to the decryption key dk satisfies the access policy $\mathbb{W}$, the algorithm outputs a message μ. Otherwise it outputs $\perp$.

Rev$(rl, id, t) \rightarrow rl$: The revocation algorithm takes as input the revocation list rl, an user's global identity id to be revoked and a time t, and outputs the updated revocation list rl.

Definition 5 (**Correctness**) The correctness of RM-CP-ABE requires that for every security parameter $\lambda \in \mathbb{N}$, every message μ, every access policy $\mathbb{W}$, and every set of attribute S which satisfy the access policy $\mathbb{W}$ it holds that

$$\Pr\left[\begin{array}{c} pp \leftarrow \text{GlobalSetup}(\lambda, N) \\ (pk_\theta, msk_\theta) \leftarrow \text{AuthSetup}(pp, \theta) \\ sk_\theta \leftarrow \text{KeyGen}(pk_\theta, msk_\theta, st, S) \\ \mu' = \mu \mid ku_t \leftarrow \text{KeyUpdate}(pk_\theta, msk_\theta, rl, st, t) \\ dk \leftarrow \text{DKGen}\big(\{sk_\theta, ku_\theta\}_{\theta \in N}\big) \\ CT \leftarrow \text{Enc}\big(pp, \{pk_\theta\}_{\theta \in N}, \mu, \mathbb{W}\big) \\ \mu' \leftarrow \text{Dec}(CT, dk) \end{array}\right] = 1$$

System model

We propose a secure CFS system framework based on the RM-CP-ABE scheme, which mainly includes four typical parties described as follows:

Attribute Authorities(**AAs**): AAs are authorization management entities set up by the enterprise side independently of the cloud server, responsible for initializing the system and broadcasting system parameters to other entities, and maintaining the access credentials of users in the system. In addition, every update cycle, the enterprise will revoke users who have lost access by publicly broadcasting key-updating materials. AAs consists of a central authority(CA) and several attribute authorities. This distributed architecture not only reduces the workload of a single attribute authority, but also enhances the scalability of the system and realizes CFS across enterprises.

Data Owners(**DOs**): DOs are the initiators and owners of shared files. It encrypts files through the client, sets corresponding access policies, and finally uploads the encrypted files to the cloud.

Data Users(**DU**): DUs are users of shared files. Users with access rights can decrypt the file through the client to read the file content.

Cloud Service Provider(**CSP**): CSP provides cloud-based file storage and file sharing services for CFS systems.

As shown in Fig. 1, the workflow of the CFS system based on the RM-CP-ABE scheme can be divided into the following four steps:

System initialization: Fig. 2 shows the CFS system initialization phase. CA runs GlobalSetup$(\lambda, N) \rightarrow pp$ and each AA runs AuthSetup$(pp, \theta) \rightarrow (pk_\theta, msk_\theta)$ to generate system parameters, then broadcast the (pp, pk_θ) to DOs and CSP(see Fig. 1 ①). AAs runs KeyGen$(pk_\theta, msk_\theta, st, S) \rightarrow sk_\theta$ to generate and send private key to the corresponding user (see Fig. 1 ②).

File sharing: As shown in Fig. 2, DOs runs Enc$(pp, \{pk_\theta\}_{\theta \in N}, \mu, \mathbb{W}) \rightarrow CT$ to encrypt the files and upload them to CSP in the file sharing phase(see Fig. 1 ⑤).

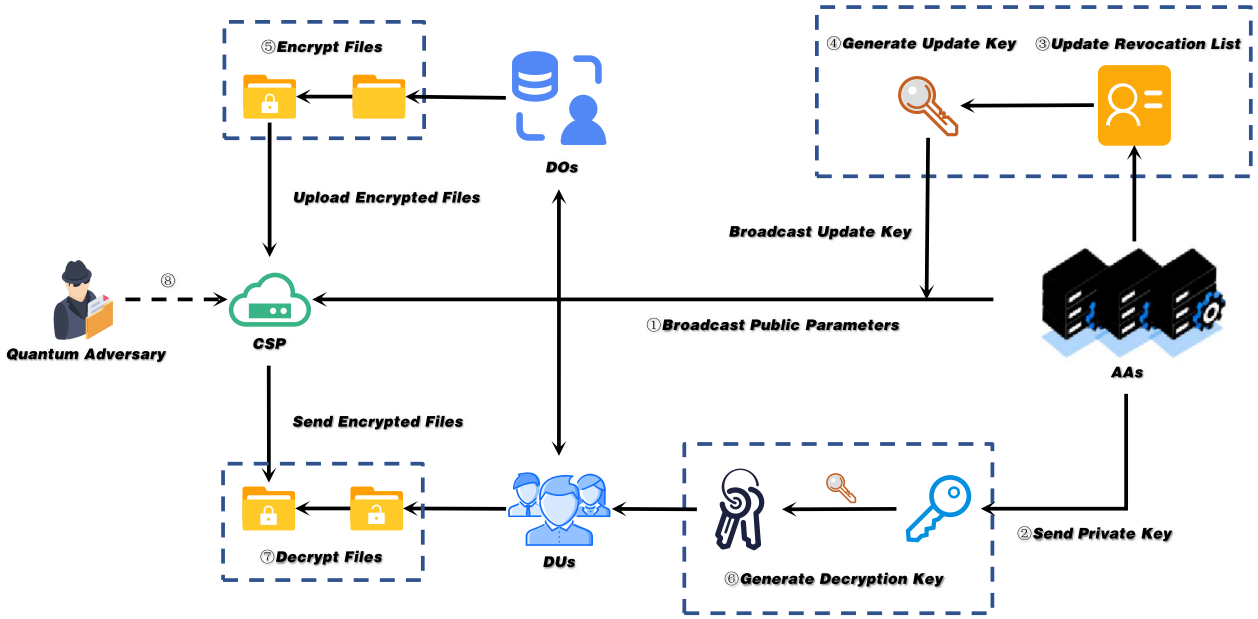

Fig. 1 System Model

User management: As shown in Fig. 3, in the user management phase, the AAs runs $\text{Rev}(rl, id, t) \rightarrow rl$ to add users who have lost access rights to the revoked list (see Fig. 1 ③), and runs algorithm $\text{KeyUpdate}(pk_\theta, msk_\theta, rl, st, t) \rightarrow ku_t$ to generate and broadcast the updated key during the key update period (see Fig. 1 ④).

Files Decrytion: As shown in Fig. 3, in the files decrytion phase, the DUs with access rights first receives the update key of the current period and runs $\text{DKGen}(\{sk_\theta, ku_\theta\}_{\theta \in N}) \rightarrow dk$ to generate a temporary decryption key(see Fig. 1 ⑥). Then DUs runs $\text{Dec}(CT, dk) \rightarrow \mu$ to decrypt and read the files contents(see Fig. 1 ⑦).

Threat model

In our system, the CSP is honest and curious, that is, it follows the protocol, but tries to get as much sensitive information in the files as possible through observations. In addition, data stored in the cloud is more vulnerable to some adversary, such as hackers. In order to prevent the semi-trusted CSP and various adversaries (including quantum adversaries, see Fig. 1⑧) from getting the sensitive information, we introduce RM-CP-ABE based on RLWE to ensure the security of CFS systems in the post-quantum era. Furthermore, in the threat model, we allow collusion attacks between revoked users and unauthorized users to attempt to obtain useful information in encrypted files. At the same time, the attacker can also

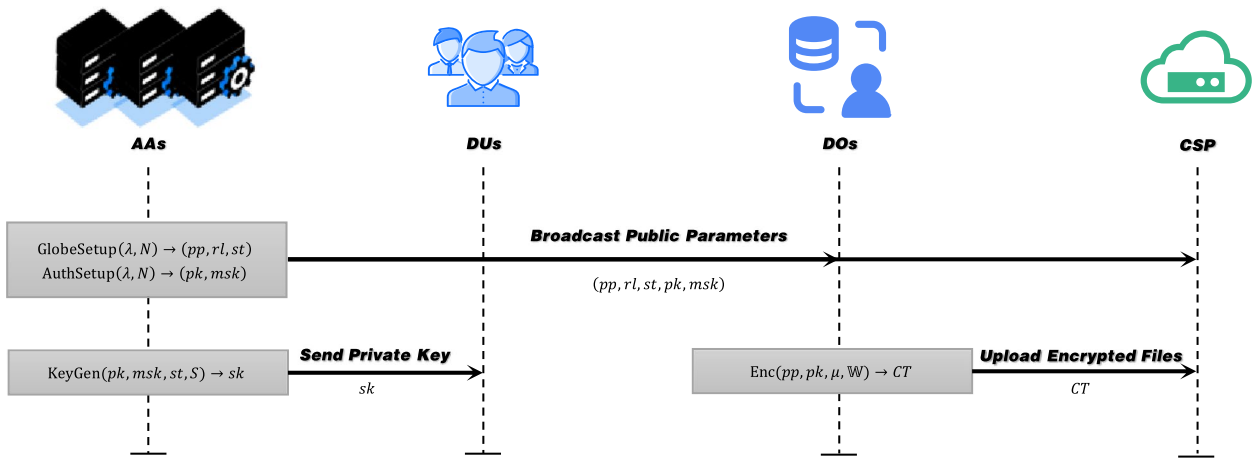

Fig. 2 System Initialization and Files sharing

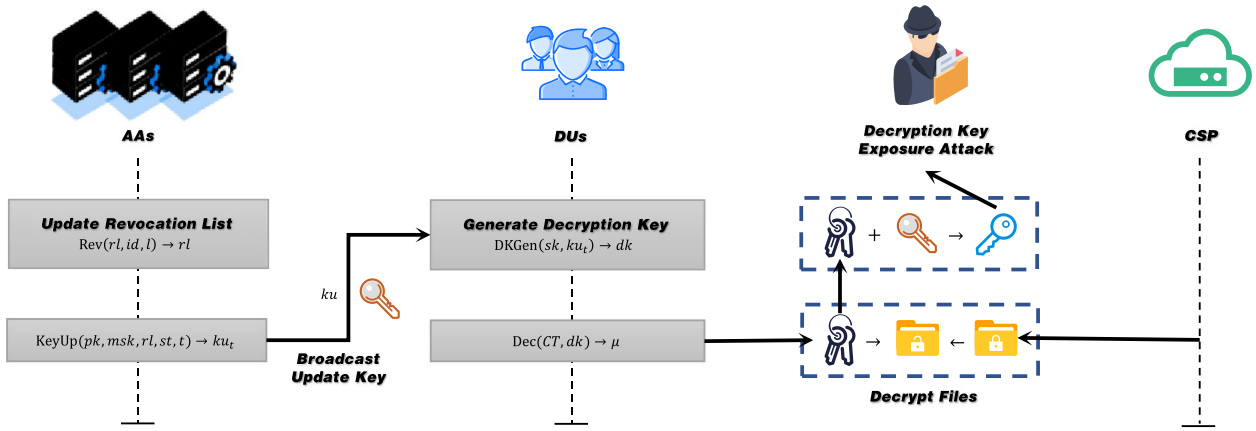

Fig. 3 User Management and Files Decryption

launch some attacks by means of the corrupted AAs. Hence, collusion attacks between the corrupted AA and the revoked users pose a serious threat to the system's security. This forces our RM-CP-ABE scheme to have the ability to resist collusion attacks. Furthermore, as shown in Fig. 3, for the practical decryption key exposure attack in indirectly revocable cryptosystems, our scheme is also required to have the ability to resist decryption key exposure. For the various types of attacks that may be encountered in the real world, we present the following security model.

Security model

Definition 6 We define the selective IND-CPA security with Decryption Key Exposure Resistance(DKER) for RM-CP-ABE scheme by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Init. The adversary $\mathcal{A}$ chooses a challenge access structure $\mathbb{W}^*$ and a time t^*, and give them to $\mathcal{C}$. The challenge access policy contains two subsets, the positive attribute set $\mathbb{W}^+$ and the negative attribute set $\mathbb{W}^-$, where $\mathbb{W}^+ \cap \mathbb{W}^- = \emptyset$. Then, $\mathcal{A}$ chooses a set of corrupted AAs **Cor** and publishes the set.

Setup. $\mathcal{C}$ calls GlobalSetup algorithm to acquire and send pp to $\mathcal{A}$. Then, $\mathcal{C}$ performs the AuthSetup algorithm to obtain and send their corresponding public and private key pairs $\{pk_\theta, msk_\theta\}_{\theta \in N}$. For the set of uncorrupted authorities, $\mathcal{C}$ sends the corresponding public key $\{pk_\theta\}_{\theta \notin \mathbf{Cor}}$. Otherwise, $\mathcal{C}$ sends the corresponding public-private key pair $\{pk_\theta, msk_\theta\}_{\theta \in \mathbf{Cor}}$.

Key Query. $\mathcal{A}$ is allowed to adaptively make the following query to $\mathcal{C}$ on the attribute set **S**:

- $sk_\theta \leftarrow \mathcal{A}^{\mathcal{O}^{KeyGen(\cdot)}}(\mathbf{S})$ for all $\theta \notin \mathbf{Cor}$.
- $ku_{\theta,t} \leftarrow \mathcal{A}^{\mathcal{O}^{KeyUpdate(\cdot)}}(t, rl, st)$ for all $\theta \notin \mathbf{Cor}$.
- $dk \leftarrow \mathcal{A}^{\mathcal{O}^{DKGen(\cdot)}}(sk_\theta, ku_{\theta,t})$.
- $(rl, st) \leftarrow \mathcal{A}^{\mathcal{O}^{Rev(\cdot)}}(rl, id, t)$.

When the adversary has access to the oracle, there are some restrictions as follows, and these restrictions are divided into two cases:

Case I. $\mathcal{O}^{KeyGen(\cdot)}$ can not be queried with the attribute set **S** satisfying $\mathbb{W}^*$.

Case II. If $\mathcal{O}^{KeyGen(\cdot)}$ was queried with the attribute set **S** satisfying $\mathbb{W}^*$, then $\mathcal{O}^{Rev(\cdot)}$ must be queried on this user id before time t^*.

In addition, both cases are held to the following restrictions:

- $\mathcal{O}^{KeyUpdate(\cdot)}$ and $\mathcal{O}^{Rev(\cdot)}$ are only allowed to query in non-decreasing time sequence.
- If $\mathcal{O}^{KeyUpdate(\cdot)}$ was queried at the time t, $\mathcal{O}^{Rev(\cdot)}$ can not be queried at the time t.
- $\mathcal{O}^{DKGen(\cdot)}$ cannot be queried for the decryption key dk corresponding to the user whose attribute set **S** satisfies $\mathbb{W}^*$ or the user that has been revoked.

Challenge. $\mathcal{A}$ chooses two equal-length message $\mu_1, \mu_2 \in \mathcal{M}$ and sends them to $\mathcal{C}$. Then $\mathcal{C}$ randomly flips a coin $b \leftarrow \{0, 1\}$ and replies $CT \leftarrow \mathrm{Enc}(pp, \{pk_\theta\}_{\theta \in N}, \mu_b, \mathbb{W})$ to $\mathcal{A}$.

Guess. The adversary $\mathcal{A}$ outputs the guessed b' for b.

Adversary $\mathcal{A}$'s advantage in this game is defined as follows:

$$Adv_{\mathcal{A}}^{IND-CPA}(\lambda) = \left|\Pr[\mathrm{b} = \mathrm{b}'] - 1/2\right|.$$

Definition 7 (**sIND − CPA in RM − CP − ABE**) An RM-CP-ABE scheme is the selective IND-CPA security with DKER, if for any PPT adversary $\mathcal{A}$, there exists

a negligible function negl(·) such that the advantage of adversary $Adv_{\mathcal{A}}^{IND-CPA}(\lambda) \leq \text{negl}(\cdot)$.

Construction

In this section, we provide the detailed construction of our RM-CP-ABE scheme for access structure represented by an AND gate over positive and negative attributes. The positive attribute in the access policy requires that the user must have this attribute to decrypt the corresponding ciphertext. On the other hand, the negative attribute in the access policy requires that the attribute set of the decrypting user cannot contain this attribute. The scheme supports N attribute authorities in the system, and efficient user revocation. The details are given below.

GlobalSetup$(\lambda, N) \rightarrow (pp, st, rl)$: The algorithm inputs a security parameter λ and the number of attribute authorities $N \in \mathbb{N}$. It first chooses a set of positive integers q, n, m. Next, it samples a vector $\mathbf{y} \in \mathcal{R}_q^{1\times m}$ and an element $\beta \in \mathcal{R}_q$. Furthermore, it specifies an FRD function H. Finally, it outputs the public parameters

$$pp = \left(q, n, m, \sigma, \sigma_s, \mathbf{y}, \beta, \mathrm{H}\right).$$

AuthSetup$(pp, \theta) \rightarrow (pk_\theta, msk_\theta)$: The algorithm takes the public parameters and the number of the attribute authority as input. Each attribute authority AA_θ runs the algorithm and generates a vector-trapdoor pair $\left(\mathbf{A}_\theta, \mathbf{T}_{\mathbf{A}_\theta}\right) \leftarrow \text{TrapGen}(\lambda)$, and a pair of vectors $\left(\mathbf{B}_{\theta,i}^{+}, \mathbf{B}_{\theta,i}^{-}\right) \leftarrow \mathcal{R}_q^{1\times m}$. Next, it samples vectors $\{\mathbf{D}_\theta, \mathbf{E}_\theta\} \leftarrow \mathcal{R}_q^{1\times m}$ and outputs the public key and master secret key for the authority θ

$$pk_\theta = \left(\mathbf{A}_\theta, \left\{\mathbf{B}_{\theta,i}^{+}; \mathbf{B}_{\theta,i}^{-}\right\}_{i\in[l_\theta]}, \mathbf{D}_\theta, \mathbf{E}_\theta\right), \quad msk_\theta = \mathbf{T}_{\mathbf{A}_\theta}.$$

KeyGen$(pk_\theta, msk_\theta, st, \mathbf{S}) \rightarrow sk$: The algorithm takes the public key pk_θ, the master secret key msk_θ of AA_θ, the state st and a user's attribute set $\mathbf{S} \subseteq l_\theta$ as input, where l_θ is the set of attributes managed by the AA_θ. It then proceeds as follows.

1 CA takes a polynomial $P(x) = \beta + \sum_{i=1}^{N-1} a_i x^i$ of degree $N-1$ for $a_i \leftarrow \mathcal{R}_q$, computes $\beta_\theta = P(\theta)$ and assigns β_θ to AA_θ.
2 For each $i \in l_\theta$, AA_θ chooses a vector $\mathbf{k}_{\theta,i} \leftarrow D_{\mathcal{R}_q^m,\sigma_s}$. If $i \in \mathbf{S}$, it computes $u_{\theta,i} \leftarrow \mathbf{B}_{\theta,i}^{+}\mathbf{k}_{\theta,i}$, else, it computes $u_{\theta,i} \leftarrow \mathbf{B}_{\theta,i}^{-}\mathbf{k}_{\theta,i}$.
3 For each node $\alpha \in Path(id) \setminus Path(id')$, if the node is empty, AA_θ randomly chooses $u_{A_\theta,\alpha,2} \leftarrow R_q$ and store it in the node, where id denotes the user's identity who generated the secret key and id' denotes the user's identity who has been revoked. It implies that the nodes $\alpha \in Path(id')$ corresponding to the revoked user identity id' have been invalidated, and the AA_θ does not need to generate secret keys corresponding to these potentially shared nodes for subsequent users.
4 AA_θ calculates $u_{A_\theta,\alpha,1} = \beta_\theta - u_{A_\theta,\alpha,2} - \sum_{i\in l_\theta} u_{\theta,i}$ and samples

$$\mathbf{k}_{\mathbf{A}_\theta,\alpha,1} \leftarrow \text{SamplePre}\left(\mathbf{A}_\theta, \mathbf{T}_{\mathbf{A}_\theta}, u_{A_\theta,\alpha,1}, \sigma, \sigma_s\right).$$

5 AA_θ samples $\mathbf{k}_{\mathbf{A}_\theta,3} \leftarrow D_{\mathcal{R}_q^{2m},\sigma_s}$ and calculates $u_{A_\theta,3} \leftarrow (\mathbf{A}_\theta \mid \mathbf{E}_\theta)\mathbf{k}_{\mathbf{A}_\theta,3}$.
6 AA_θ chooses $u_{A_\theta,4} \leftarrow \mathcal{R}_q$ and stores it.
7 Then AA_θ samples

$$\mathbf{k}_{\mathbf{A}_\theta,4} \leftarrow \text{SamplePre}\left(\mathbf{A}_\theta, \mathbf{T}_{\mathbf{A}_\theta}, u_{A_\theta,4} - u_{A_\theta,3}, \sigma, \sigma_s\right)$$

.
8 Finally, it outputs the secret key

$$sk_\theta = \left(\left\{\mathbf{k}_{\mathbf{A}_\theta,\alpha,1}\right\}_{\alpha\in Path(id)\setminus Path(id')}, \left\{\mathbf{k}_{\theta,i}\right\}_{i\in S}, \mathbf{k}_{\mathbf{A}_\theta,3}, \mathbf{k}_{\mathbf{A}_\theta,4}\right)$$

KeyUpdate$(pk_\theta, msk_\theta, rl, st, t) \rightarrow ku$: Given the public key pk_θ, the master secret key msk_θ of AA_θ, the revocation list rl, the state st and a revocation time t, the algorithm proceeds as follows.

1 It samples $r_t \leftarrow D_{\mathcal{R}_q,\sigma_s}$ and samples $\mathbf{k}_{\mathbf{A}_\theta,5} \leftarrow D_{\mathcal{R}_q^{2m},\sigma_s}$.
2 It calculates $u_{A_\theta,5} = (\mathbf{A}_\theta \mid \mathbf{E}_\theta)\mathbf{k}_{\mathbf{A}_\theta,5}$.
3 Next, it fetches $u_{A_\theta,\alpha,2}$ and $u_{A_\theta,4}$, and samples

$$\mathbf{k}_{\mathbf{A}_\theta,\alpha,2} \leftarrow \text{SampleLeft}\left(\mathbf{A}_\theta, \mathbf{D}_\theta + \mathbf{H}_t, \mathbf{T}_{\mathbf{A}_\theta}, u'_{A_\theta,\alpha,2}\right)$$

where $\alpha \in \text{KUNodes}(st, rl, t)$, $u'_{A_\theta,\alpha,2} = u_{A_\theta,\alpha,2} - u_{A_\theta,4}r_t + u_{A_\theta,5}$ and $\mathbf{H}_t = Tran_{M\rightarrow V}\left(Tran_{V\rightarrow M}\left(\mathbf{y}^\top\right)\mathrm{H(t)}\right)$.
4 Finally, it outputs the key update

$$ku_{\theta,t} = \left(\left\{\mathbf{k}_{\mathbf{A}_\theta,\alpha,2}\right\}_{\alpha\in\text{KUNodes}(st,rl,t)}, \mathbf{k}_{\mathbf{A}_\theta,5}, r_t\right).$$

DKGen$(\{sk_\theta, ku_\theta\}_{\theta\in N}) \rightarrow dk$: The algorithm takes the secret key $\{sk_\theta\}_{\theta\in N}$ and the key update $\{ku_\theta\}_{\theta\in N}$ as input. If $Path(id) \cap \text{KUNodes}(st, rl, t) = \emptyset$, it returns a failure symbol $\perp$. Otherwise, it can find a unique node $\alpha \in Path(id) \cap \text{KUNodes}(st, rl, t)$ and store $\left(\mathbf{k}_{\mathbf{A}_\theta,\alpha,1}, \mathbf{k}_{\mathbf{A}_\theta,\alpha,2}, \mathbf{k}_{\theta,i}\right)$ in dk_θ for ench $\theta \in N$. For simplicity, we can omit the subscript α. Then, let $\mathbf{dk}_{\mathbf{A}_\theta,1} = \mathbf{k}_{\mathbf{A}_\theta,1} + \mathbf{k}_{\mathbf{A}_\theta,4}r_t$, $\mathbf{dk}_{\mathbf{A}_\theta,2} = \mathbf{k}_{\mathbf{A}_\theta,2}$, and $\mathbf{dk}_{\mathbf{A}_\theta,3} = \mathbf{k}_{\mathbf{A}_\theta,3}r_t - \mathbf{k}_{\mathbf{A}_\theta,5}$, Finally, it outputs the decryption key

$$dk = \left\{\mathbf{dk}_{\mathbf{A}_\theta,1}, \mathbf{dk}_{\mathbf{A}_\theta,2}, \mathbf{dk}_{\mathbf{A}_\theta,3}, \left\{\mathbf{k}_{\theta,i}\right\}_{i\in S}\right\}_{\theta\in N}.$$

Enc$(pp, \{pk_\theta\}_{\theta\in N}, \mu, \mathbb{W}) \rightarrow CT$: Given the public parameters pp, the public key $\{pk_\theta\}_{\theta\in N}$, a message μ,

and an access structure $\mathbb{W} = (\mathbb{W}^+ \cap \mathbb{W}^-)$, which determines the set of positive and negative attributes, the algorithm samples $s \leftarrow \mathcal{R}_q$ and $e_0 \leftarrow D_{\mathcal{R}_q,\sigma_s}$, and computes $\mathbf{C}_0 \leftarrow s\beta + e_0 + \mu[q/2]$. Then it samples vectors $\{\mathbf{e}_{\mathbf{A}_\theta}\}_{\theta\in N} \leftarrow D_{\mathcal{R}_q^m,\sigma_s}$, and computes $\mathbf{C}_{\mathbf{A}_\theta} \leftarrow \mathbf{A}_\theta^T s + \mathbf{e}_{\mathbf{A}_\theta}$. For each $\theta \in N$ and $i \in l_\theta$, if $i \in \mathbb{W}^+$, it samples vectors $\mathbf{e}_{\theta,i} \leftarrow D_{\mathcal{R}_q^m,\sigma_s}$ and computes $\mathbf{C}_{\theta,i} \leftarrow (\mathbf{B}_{\theta,i}^+)^T s + \mathbf{e}_{\theta,i}$, else if $i \in \mathbb{W}^-$, it samples vectors $\mathbf{e}_{\theta,i} \leftarrow D_{\mathcal{R}_q^m,\sigma_s}$ and computes $\mathbf{C}_{\theta,i} \leftarrow (\mathbf{B}_{\theta,i}^-)^T s + \mathbf{e}_{\theta,i}$. Otherwise, it samples vectors $\{\mathbf{e}_{\theta,i}^+, \mathbf{e}_{\theta,i}^-\} \leftarrow D_{\mathcal{R}_q^m,\sigma_s}$, and computes $\mathbf{C}_{\theta,i}^+ \leftarrow (\mathbf{B}_{\theta,i}^+)^T s + \mathbf{e}_{\theta,i}^+$ and $\mathbf{C}_{\theta,i}^- \leftarrow (\mathbf{B}_{\theta,i}^-)^T s + \mathbf{e}_{\theta,i}^-$. Next, it samples matrices $\{\mathbf{R}_{\theta,\mathbf{1}}, \mathbf{R}_{\theta,\mathbf{2}}\} \leftarrow \{\pm 1\}^{m\times m}$, and computes $\mathbf{C}_{\theta,t,1} = (\mathbf{D}_\theta + \mathbf{H}_t)^T s + \mathbf{R}_{\theta,1}\mathbf{e}_{\mathbf{A}_\theta}$ and $\mathbf{C}_{\theta,t,2} = (\mathbf{E}_\theta)^T s + \mathbf{R}_{\theta,2}\mathbf{e}_{\mathbf{A}_\theta}$. Finally, it outputs the ciphertext

$$CT = \left(\mathbf{C}_0, \{\mathbf{C}_{\mathbf{A}_\theta}, \mathbf{C}_{\theta,t,1}, \mathbf{C}_{\theta,t,2}\}_{\theta\in N}, \left\{\mathbf{C}_{\theta,i}, \mathbf{C}_{\theta,i}^{\pm}\right\}_{\theta\in N, i\in l_\theta}\right).$$

$\mathrm{Dec}(CT, dk) \rightarrow \mu$: The algorithm takes the ciphertext CT and the decryption key dk as input. Let $\mathbf{S}$ be the attribute set associated to dk. If $\mathbf{S} \cap \mathbb{W}^+ = \mathbb{W}^+$ and $\mathbf{S} \cap \mathbb{W}^- = \emptyset$, it proceeds as follows:

1 Calculate $a_{\theta,1} = (\mathbf{C}_{\mathbf{A}_\theta})^\top \mathbf{dk}_{\mathbf{A}_\theta,1}$.
2 For each $\theta \in N$ and $i \in l_\theta$, if $i \in \mathbb{W}$, calculate $a_{\theta,2,i} = (\mathbf{C}_{\theta,i})^\top \mathbf{k}_{\theta,i}$, else if $i \in S$, calculate $a_{\theta,2,i} = (\mathbf{C}_{\theta,i}^+)^\top \mathbf{k}_{\theta,i}$, otherwise, calculate $a_{\theta,2,i} = (\mathbf{C}_{\theta,i}^-)^\top \mathbf{k}_{\theta,i}$.
3 Let $\mathbf{C}_{\mathbf{A}_\theta,t,1} = (\mathbf{A}_\theta \mid \mathbf{D}_\theta + \mathbf{H}_t)^T s + (\mathbf{e}_{\mathbf{A}_\theta} \mid \mathbf{R}_{\theta,1}\mathbf{e}_{\mathbf{A}_\theta})$, and calculate $a_{\theta,3} = \mathbf{C}_{\mathbf{A}_\theta,t}\mathbf{dk}_{\mathbf{A}_\theta,2}$.
4 Let $\mathbf{C}_{\mathbf{A}_\theta,t,2} = (\mathbf{A}_\theta \mid \mathbf{E}_\theta)^T s + (\mathbf{e}_{\mathbf{A}_\theta} \mid \mathbf{R}_{\theta,2}\mathbf{e}_{\mathbf{A}_\theta})$, and calculate $a_{\theta,4} = \mathbf{C}_{\mathbf{A}_\theta,t}\mathbf{dk}_{\mathbf{A}_\theta,3}$.
5 Calculate

$$t = \mathbf{C}_0 - \sum_{\theta\in N} \mathcal{L}_\theta\left(a_{\theta,1} + a_{\theta,2} + a_{\theta,3} + a_{\theta,4}\right)$$

where $a_{\theta,2} = \sum_{i\in l_\theta} a_{\theta,2,i}$, and $\mathcal{L}_\theta = \frac{\Pi_{\theta\in N,\theta\neq\delta}(-\theta)}{\Pi_{\theta\in N,\theta\neq\delta}(\delta-\theta)}$ is the Lagrangian coefficient.
6 If $|t_i| < \frac{q}{4}$, output $\mu_i = 0$, otherwise output $\mu_i = 1$.

Otherwise, it outputs a failure symbol $\perp$.

$\mathrm{Rev}(rl, id, t) \rightarrow rl$: The algorithm takes as input the revocation list rl, an user's global identity id to be revoked and a time t. It updates revocation list rl

$$rl \leftarrow rl \cup (id, t).$$

Correctness

We assume that an attribute set $\mathbf{S}$ satisfies the access policy($\mathbf{S} \cap \mathbb{W}^+ = \mathbb{W}^+$ and $\mathbf{S} \cap \mathbb{W}^- = \emptyset$), then we have

$$\begin{aligned}
&a_{\theta,1} + a_{\theta,2} + a_{\theta,3} + a_{\theta,4}\\
=&(\mathbf{C}_{\mathbf{A}_\theta})^\top \mathbf{dk}_{\mathbf{A}_\theta,1} + \sum_{i\in l_\theta}\left(\mathbf{C}_{\theta,i}^*\right)^\top \mathbf{k}_{\theta,i} + (\mathbf{C}_{\mathbf{A}_\theta,t,1})^\top \mathbf{dk}_{\mathbf{A}_\theta,2}\\
&+ (\mathbf{C}_{\mathbf{A}_\theta,t,2})^\top \mathbf{dk}_{\mathbf{A}_\theta,3}\\
=&s\mathbf{A}_\theta \mathbf{dk}_{\mathbf{A}_\theta,1} + \sum_{i\in l_\theta} s\mathbf{B}_{\theta,i}^* \mathbf{k}_{\theta,i} + s(\mathbf{A}_\theta \mid \mathbf{D}_\theta + \mathbf{H}_t)\mathbf{dk}_{\mathbf{A}_\theta,2}\\
&+ s(\mathbf{A}_\theta \mid \mathbf{E}_\theta)\mathbf{dk}_{\mathbf{A}_\theta,3} + \mathbf{e}_{\mathbf{A}_\theta}^\top \mathbf{dk}_{\mathbf{A}_\theta,1} + \mathbf{e}_{\theta,i}^\top \mathbf{k}_{\theta,i}\\
&+ (\mathbf{e}_{\mathbf{A}_\theta} \mid \mathbf{R}_{\theta,1}\mathbf{e}_{\mathbf{A}_\theta})^\top \mathbf{dk}_{\mathbf{A}_\theta,2} + (\mathbf{e}_{\mathbf{A}_\theta} \mid \mathbf{R}_{\theta,2}\mathbf{e}_{\mathbf{A}_\theta})^\top \mathbf{dk}_{\mathbf{A}_\theta,3}\\
=&s\mathbf{A}_\theta \mathbf{k}_{\mathbf{A}_\theta,1} + s\mathbf{A}_\theta \mathbf{k}_{\mathbf{A}_\theta,4} r_t + \sum_{i\in l_\theta} s\mathbf{B}_{\theta,i}^* \mathbf{k}_{\theta,i} + s(\mathbf{A}_\theta \mid \mathbf{E}_\theta)\mathbf{k}_{\mathbf{A}_\theta,3} r_t\\
&+ s(\mathbf{A}_\theta \mid \mathbf{D}_\theta + \mathbf{H}_t)\mathbf{k}_{\mathbf{A}_\theta,2} - s(\mathbf{A}_\theta \mid \mathbf{E}_\theta)\mathbf{k}_{\mathbf{A}_\theta,5} + \widetilde{\mathbf{e}}\\
=&su_{A_\theta,1} + (su_{A_\theta,4} - su_{A_\theta,3})r_t + \sum_{i\in l_\theta} su_{\theta,i} + s\left(u'_{A_\theta,\alpha,2}\right)\\
&+ su_{A_\theta,3} r_t - su_{A_\theta,5} + \widetilde{\mathbf{e}}_\theta\\
=&s\beta_\theta + \widetilde{\mathbf{e}}_\theta
\end{aligned}$$

where, if $i \in \mathbb{W}$, $\mathbf{C}_{\theta,i}^* = \mathbf{C}_{\theta,i}$, else if $i \in S$, $\mathbf{C}_{\theta,i}^* = \mathbf{C}_{\theta,i}^+$, otherwise, $\mathbf{C}_{\theta,i}^* = \mathbf{C}_{\theta,i}^-$, and the choice of $\mathbf{B}_{\theta,i}^*$ corresponds to that of $\mathbf{C}_{\theta,i}^*$. Furthermore, the total noise term is denoted by $\widetilde{\mathbf{e}}_\theta = \mathbf{e}_{\mathbf{A}_\theta}^\top \mathbf{k}_{\mathbf{A}_\theta,1} + \mathbf{e}_{\mathbf{A}_\theta}^\top \mathbf{k}_{\mathbf{A}_\theta,4} r_t + \mathbf{e}_{\theta,i}^\top \mathbf{k}_{\theta,i} + (\mathbf{e}_{\mathbf{A}_\theta} \mid \mathbf{R}_{\theta,1}\mathbf{e}_{\mathbf{A}_\theta})^\top \mathbf{k}_{\mathbf{A}_\theta,2} + (\mathbf{e}_{\mathbf{A}_\theta} \mid \mathbf{R}_{\theta,2}\mathbf{e}_{\mathbf{A}_\theta})^\top (\mathbf{k}_{\mathbf{A}_\theta,3} r_t + \mathbf{k}_{\mathbf{A}_\theta,5})$. Next, for $\forall\theta \in N$, we compute

$$\begin{aligned}
&\mathbf{C}_0 - \sum_{\theta\in N} \mathcal{L}_\theta\left(a_{\theta,1} + a_{\theta,2} + a_{\theta,3} + a_{\theta,4}\right)\\
=&s\beta + e_0 + \mu[q/2] - \sum_{\theta\in N} \mathcal{L}_\theta\left(s\beta_\theta + \widetilde{\mathbf{e}}_\theta\right)\\
=&\mu[q/2] + e_0 - \sum_{\theta\in N} \mathcal{L}_\theta \widetilde{\mathbf{e}}_\theta.
\end{aligned}$$

When the noise is small enough, it will not affect the plaintext information in the ciphertext. We let the upper bound on the combination of all noise factors be η, the upper bounds of key components $\{\mathbf{k}_{\mathbf{A}_\theta,1}, \mathbf{k}_{\mathbf{A}_\theta,2}, \mathbf{k}_{\mathbf{A}_\theta,3}, \mathbf{k}_{\mathbf{A}_\theta,4}, \mathbf{k}_{\mathbf{A}_\theta,5}, \mathbf{k}_{\theta,i}\}$ are η_s, and the upper bounds of ciphertext noise factors $\mathbf{e}_{\mathbf{A}_\theta}$, $\mathbf{e}_{\theta,i}$ are η_e. According to [19], we let $\eta_e = 8\sigma$, and $\eta_s = 8\sigma_s$. In order to ensure the correctness of decryption, the following inequality holds with non-negligible probability, i.e.,

$$\eta = \eta_e \eta_s \sqrt{nm(l + (3 + 2\eta_e)N)} < \frac{q}{4},$$

where l is the sum of the number of attributes in all attribute authorities. Finally, we have that

$$q > 256\sigma\sigma_s \sqrt{nm(l + (3 + 16\sigma)N)}.$$

Security proof

Theorem 1 *If the RLWE assumption holds, then our scheme is secure against the selective IND-CPA described in Security model section.*

In the security proof, we divided the adversary into two types, one is the adversary who has not obtained legal authorization, and the user's private key they have does not satisfies the challenge access policy $\mathbb{W}^*$. The other category is revoked users who indicate malicious intent. Adversary can query the user's private key that satisfies the challenge access policy and the updated key and decryption key before the revoked time t ($t < t^*$). To prove the above theorem, we will describe several security games that differ from each other in the formation of public parameters, the key queried by adversary $\mathcal{A}$ and the challenge ciphertext. The first game is the same as the ABE game we defined in the security model, and the adversary's advantage is zero in the last game of the sequence. And we argue that the adversary $\mathcal{A}$'s advantage varies negligibly between each successive security game. This will prove that the adversary has a negligible advantage in winning the original ABE security game.

Game$_0$: This is the real selective security game form Security model section between an adversary $\mathcal{A}$ against our scheme and a RM-CP-ABE challenger $\mathcal{B}$.

Game$_1$: This game is the same as the previous game except the way the public key vectors $\{\mathbf{A}_\theta, \mathbf{B}^{\pm}_{\theta,i}, \mathbf{D}_\theta, \}$ are generated for all $\theta \in N$ and $i \in l_\theta$ during the setup phase. In this game, the public key vectors $\{\mathbf{A}_\theta\}_{\theta\in N}$ is uniformly randomly chosen over $\mathcal{R}_q^{1\times m}$ instead of by the TrapGen algorithm. For all $\theta \in N$ and $i \in l_\theta \setminus \mathbb{W}^*$, $\mathcal{B}$ samples $\{\mathbf{B}^+_{\theta,i}, \mathbf{B}^-_{\theta,i}\} \leftarrow \mathcal{R}_q^{1\times m}$. For each $i \in \mathbb{W}^+$, $\mathcal{B}$ samples $\mathbf{B}^+_{\theta,i} \leftarrow \mathcal{R}_q^{1\times m}$ and computes $\left\{\mathbf{B}^-_{\theta,i}, \mathbf{T}^-_{\mathbf{B}_{\theta,i}}\right\} \leftarrow \text{TrapGen}(\lambda)$. Correspondingly, $\mathcal{B}$ samples $\mathbf{B}^-_{\theta,i} \leftarrow \mathcal{R}_q^{1\times m}$ and computes $\left\{\mathbf{B}^+_{\theta,i}, \mathbf{T}^+_{\mathbf{B}_{\theta,i}}\right\} \leftarrow \text{TrapGen}(\lambda)$ for each $i \in \mathbb{W}^-$. Then, $\mathcal{B}$ samples $\{\mathbf{R}_{\theta,1}, \mathbf{R}_{\theta,2}\} \leftarrow \{-1, 1\}^{m\times m}$ for each $\theta \in N$. Next, it computes $\mathbf{D}_\theta \leftarrow \mathbf{A}_\theta\mathbf{R}_{\theta,1} - \mathbf{H}_{t^*}$ and $\mathbf{E}_\theta \leftarrow \mathbf{A}_\theta\mathbf{R}_{\theta,2}$. According to the properties of the TrapGen algorithm and the leftover hash lemma, we conclude that **Game$_0$** and **Game$_1$** are statistically indistinguishable from the adversary's view.

Game$_2$: This game is analogous to the previous game except the generationed of the public parameter $\mathbf{y}$ during the setup phase. In this game, the vector $\mathbf{y}$ is generated by TrapGen(λ) instead of being randomly sampled. The indistinguishably between **Game$_1$** and **Game$_2$** follows from the good properties of the TrapGen($\cdot$) algorithm.

Game$_3$: In this game, we change the way the secret keys $\{sk_\theta\}_{\theta\in N}$ and update keys $\{ku_{\theta,t}\}_{\theta\in N}$ are generated during the global setup phase. According to the key query restrictions on the two cases in the security model, we make the following changes:

Case I: If the adversary $\mathcal{A}$ query an identity *id* whose attribute set $\mathbf{S}$ does not satisfy the access policy $\mathbb{W}^*$, for all $\theta \in N$ and $t = t^*$, $\mathcal{B}$ chooses $\{\mathbf{k}_{\mathbf{A}_\theta,\alpha,2}, \mathbf{k}_{\mathbf{A}_\theta,4}\} \leftarrow \mathcal{R}_q^m$ and computes $u_{A_\theta,4} = \mathbf{A}_\theta\mathbf{k}_{\mathbf{A}_\theta,4} + u_{A_\theta,3}$. Then it computes $u_{A_\theta,\alpha,2} = (\mathbf{A}_\theta \mid \mathbf{A}_\theta\mathbf{R}_{\theta,1})\mathbf{k}_{\mathbf{A}_\theta,\alpha,2} + u_{A_\theta,4}r_{t^*} - u_{A_\theta,5}$ and store them in node. Next, for any $t \neq t^*$, $\mathcal{B}$ samples $\mathbf{k}_{\mathbf{A}_\theta,\alpha,2} \leftarrow \text{SampleRight}\,(\mathbf{A}_\theta, \mathbf{A}_\theta\mathbf{R}_{\theta,1} - \mathbf{H}_{t^*} + \mathbf{H}_t, \mathbf{T}_{\mathbf{A}_\theta}, u_{A_\theta,\alpha,2} - u_{A_\theta,4}(r_t - r_{t^*}) + u_{A_\theta,5})$. For secret key query, $\mathcal{B}$ chooses $\mathbf{k}_{\mathbf{A}_\theta,\alpha,1} \leftarrow \mathcal{R}_q^{1\times m}$ and computes $u_{A_\theta,\alpha,1} = \mathbf{A}_\theta\mathbf{k}_{\mathbf{A}_\theta,\alpha,1}$ for all $\theta \in N$. Since $\mathbf{S}$ does not satisfy $\mathbb{W}^*$, $\mathcal{B}$ must knows at least one $\mathbf{T}^+_{\mathbf{B}_{\theta,j}}$ or $\mathbf{T}^-_{\mathbf{B}_{\theta,j}}$. Then, $\mathcal{B}$ calculates $u_{\theta,i} \leftarrow \mathbf{B}^-_{\theta,i}\mathbf{k}_{\theta,i}$ for each $i \in \mathbf{S}$, $i \neq j$ and samples $\mathbf{k}_{\theta,j} \leftarrow \text{SamplePre}\left(\mathbf{B}^*_{\theta,j}, \mathbf{T}^*_{\mathbf{B}_{\theta,\mathbf{j}}}, u_{\theta,j}, \sigma, \sigma_s\right)$, where $u_{\theta,j} = \beta_\theta - u_{A_\theta,\alpha,1} - u_{A_\theta,\alpha,2} - \sum_{i\in l_\theta, i\neq j} u_{\theta,i}$.

Case II: If the adversary $\mathcal{A}$ query an identity *id* whose attribute set $\mathbf{S}$ does satisfy the access policy $\mathbb{W}^*$, then the identity *id* have been revoked before a time t^*. For each $\theta \in N$ and $\alpha \in Path(id) \cap \text{KUNodes}(st, rl, t)$, $\mathcal{B}$ chooses $\mathbf{k}_{\mathbf{A}_\theta,\alpha,1} \leftarrow \mathcal{R}_q^m$ and computes $u_{A_\theta,\alpha,1} = \mathbf{A}_\theta\mathbf{k}_{\mathbf{A}_\theta,\alpha,1}$. Then, $\mathcal{B}$ computes $u_{A_\theta,\alpha,2} = \beta_\theta - u_{A_\theta,\alpha,1} - \sum_{i\in l_\theta} u_{\theta,i}$ and store them in the node.

The indistinguishably between **Game$_2$** and **Game$_3$** follows from the good sampling property of SamplePre($\cdot$) algorithm.

Game$_4$: This game is identical to **Game$_3$**, except that the challenge ciphertext is generated. In this game, for all $\theta \in N$, $i \in l_\theta \setminus \mathbb{W}^*$, the challenger $\mathcal{B}$ sets $\{\mathbf{C}_{\mathbf{A}_\theta}, \mathbf{C}_{\theta,i}, \mathbf{C}^{\pm}_{\theta,i}\} \leftarrow \mathcal{R}_q^m$, $\mathbf{C}_{\theta,t,1} = \mathbf{R}_{\theta,1}\mathbf{C}_{\mathbf{A}_\theta}$, and $\mathbf{C}_{\theta,t,2} = \mathbf{R}_{\theta,2}\mathbf{C}_{\mathbf{A}_\theta}$. The indistinguishably between **Game$_3$** and **Game$_4$** follows from the RLWE problem. Since the challenge ciphertext is a random element in the ciphertext space, the advantage of adversary $\mathcal{A}$ in this game is negligible.

Efficiency analysis

In this section, we carry out theoretical analysis and simulation implementation of our scheme. In terms of theoretical analysis, we compare our scheme and related work in terms of scheme characteristics, storage cost, and computational cost. In the experimental simulation, we focus on analyzing the time cost of each algorithm when the number of attributes is different.

Theoretical analysis

In this subsection, we gave a theoretical analysis in functions of schemes, storage cost, and computational cost by comparing related schemes in [8, 9, 12, 13, 17, 18, 25] with our scheme.

Table 1 shows the comparison of related schemes and our scheme in terms of security assumptions and the scheme's functions. Observe that schemes in [9, 17, 18, 25]

Table 1 Scheme Functions

Scheme	Model	Multi-authority	Revocable	DKER	Assumption
Scheme [17]	CP-ABE	×	×	×	LWE
Scheme [13]	CP-ABE	✓	×	×	LWE
Scheme [12]	CP-ABE	✓	×	×	LWE
Scheme [9]	CP-ABE	×	✓	×	LWE
Scheme [25]	KP-ABE	×	✓	✓	LWE
Scheme [18]	CP-ABE	×	×	×	RLWE
Scheme [8]	CP-ABE	✓	✓	×	RLWE
Our Scheme	CP-ABE	✓	✓	✓	RLWE

only support single attribute authority. While schemes in [8, 12, 13] and our scheme support multi-attribute authority, which can effectively reduce the computational burden of each attribute authority. In addition, the schemes in [8, 9, 25] and our scheme also support user revocation, allowing the system to manage users dynamically, improving the practicability of the scheme. On this basis, the scheme in [25] and our scheme also considers more possible security issues with decryption key exposure resistance.

Table 2 shows the storage cost of related schemes and our scheme in terms of public parameter size, private key size, and ciphertext size. For the convenience of description, we let N denote the number of attribute authorities in the system, $|l_s|$ denote the total number of attributes in the system, $|l_a|$ denote the number of attributes in the access policy, and $|l_u|$ denote the number of attributes owned by users. It is worth mentioning that schemes in [9, 12, 13, 17, 25] are constructed based on the LWE assumption, where the public parameter $m = \Omega(n \log q)$, while the scheme in [8, 18] and our scheme are constructed based on the RLWE assumption. It is well-known that the scheme based on RLWE is usually superior to the schemes based on LWE in the parameter size and efficiency. Therefore, the scheme in [8, 18] and our scheme have smaller parameters than schemes [9, 12, 13, 17, 25], and are more efficient.

Experimental simulation

Test environment

We have implemented our construction on an Ubuntu 18.04 operating system with Intel Core i5-10400F, 2.90GHz processor and 4GB of memory with the Palisade library.

Storage cost

In our implementation, we use the Gaussian sampling algorithm for rings in scheme [19]. Specifically, we set the base $b = 64$, the ring size $n = 1024$ and the number of attribute authority $N = 3$. Table 3 shows the storage overhead of our scheme under the above parameter settings and different number of attributes. Note that the ciphertext size of our scheme will change according to $|l_a|$. When $|l_a| = 1$, the ciphertext size reaches the maximum value, the formula for ciphertext size (in number of bits) can be given as $(2|l_s| + 2N) \cdot m \cdot n \cdot \lceil \log q \rceil$. When $|l_a| = |l_s|$, the ciphertext size reaches the minimum value, the formula can be given as $(|l_s| + 2N + 1) \cdot m \cdot n \cdot \lceil \log q \rceil$. Compared with the schemes based on the LWE assumption, our scheme has a relatively small cost in terms of storage. Moreover, as the number of attributes in the system increases, the public key size, decryption key size and ciphertext size of our scheme increase slowly. Therefore, our scheme is completely feasible in practical scenarios.

Situation results

Figure 4 shows the time cost comparison between recent work [8] (refer to YSL) and our scheme in initialization phase, private key generation phase, update key generation phase and encryption phase. We set the number of attributes vary from 6 to 18 in an increment of 2, hence there are 7 different situations. For each case, the experiment was repeated 20 times and each experiment was completely independent, and finally the average value was taken as the experimental result. In Fig. 4a, since YSL and our scheme both support multiple attribute authorities to work together, the time cost in the system

Table 2 Storage cost comparison

Scheme	Public parameter size	Decryption key size	Ciphertext size
Scheme [17]	$(2\|l_s\|+2)mn\lceil\log q\rceil$	$2(\|l_u\|+\|l_s\|)mn\lceil\log q\rceil$	$(2\|l_a\|+2\|l_s\|+1)mn\lceil\log q\rceil$
Scheme [13]	$\|l_s\|(m+2)n\lceil\log q\rceil$	$\|l_u\|m\lceil\log q\rceil$	$(\|l_a\|+1)m\lceil\log q\rceil$
Scheme [12]	$2\|l_s\|mn\lceil\log q\rceil$	$2\|l_u\|m\lceil\log q\rceil$	$(2\|l_a\|+1)m\lceil\log q\rceil$
Scheme [9]	$5\|l_s\|(m+1)n\lceil\log q\rceil$	$4\|l_s\|m\lceil\log q\rceil$	$(3\|l_s\|m+1)\lceil\log q\rceil$
Scheme [25]	$(\|l_s\|+3)mn\lceil\log q\rceil$	$3mn\lceil\log q\rceil$	$(\|l_s\|+2)m\lceil\log q\rceil$
Scheme [18]	$(2\|l_s\|+1)mn\lceil\log q\rceil$	$(\|l_u\|+\|l_s\|)mn\lceil\log q\rceil$	$(2\|l_a\|+2\|l_s\|)mn\lceil\log q\rceil$
Scheme [8]	$(4\|l_s\|+N+2)mn\lceil\log q\rceil$	$4(\|l_u\|+\|l_s\|)mn\lceil\log q\rceil$	$(2\|l_a\|+2\|l_s\|+1)mn\lceil\log q\rceil$
Our Scheme	$(\|l_s\|+3N+1)mn\lceil\log q\rceil$	$(\|l_s\|+3N)mn\lceil\log q\rceil$	$\leq (2\|l_s\|+2N)mn\lceil\log q\rceil$

Table 3 Storage cost of our scheme

(l_s, l_a)	Public parameter size	Decryption key size	Ciphertext size
(6, 1)	2.67 MB	2.5 MB	3 MB
(6, 6)	2.67 MB	2.5 MB	2.17 MB
(12, 1)	3.67 MB	3.5 MB	6 MB
(12, 12)	3.67 MB	3.5 MB	3.17 MB
(18, 1)	4.67 MB	4.5 MB	8 MB
(18, 18)	4.67 MB	4.5 MB	4.17 MB

initialization phase is both small. And both increase linearly with the increase of the number of attributes, and the growth rate is about 1.42 (ms/item). Therefore, even in a system with a large number of attributes, the setup phase time cost is acceptable. Figure 4b shows the time cost of YSL and our scheme in terms of user private key generation. Since our scheme has smaller user key size and number of key components than YSL, it takes less time. And as the number of attributes increases, the time cost of our scheme grows more slowly. In Fig. 4c, YSL has a slightly smaller time cost in updating key generation than our scheme. However, the efficiency of update key generation is related to the number of users in the system and not to the number of attributes. In YSL, the time cost of generating updated keys is linear with the number of users in the system, while the time cost of our scheme grows logarithmically with the number of users in the system. Therefore, in the case of a large number of attributes and users in the system, our update key generation algorithm can still maintain a high efficiency. Figure 4d shows the time cost comparison between YSL and our scheme in terms of encryption. During the encryption process, our scheme operates more compactly, and the number of ciphertext components associated with each attribute is also less than YSL. Therefore, the time cost of our scheme in the encryption phase is less than that of YSL, and as the number of attributes increases, the time cost of our scheme increases more slowly.

Figure 5 shows the relationship between the number of users in the system and the generation time of the update key. We denote the number of users by N_u. When no user in the system has been revoked, all users share a set of update keys. At this time, the attribute authority only needs to generate a set of update keys. When a user is revoked, the attribute authority needs to generate $\lceil \log N_u \rceil$ update key components. Observe that the time cost of updating the key generation algorithm grows linearly and slowly when the number of users in the system grows exponentially. Therefore, our scheme is also suitable for large systems with a large number of users.

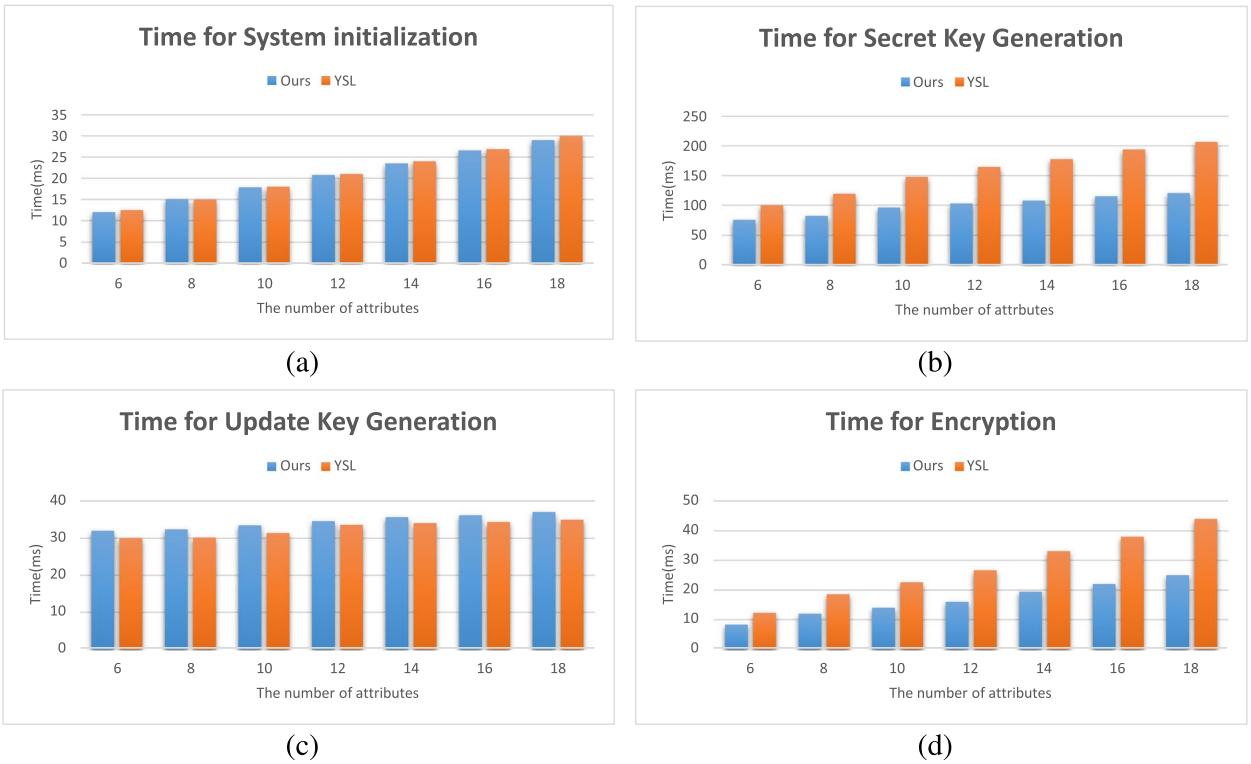

Fig. 4 Time cost of scheme

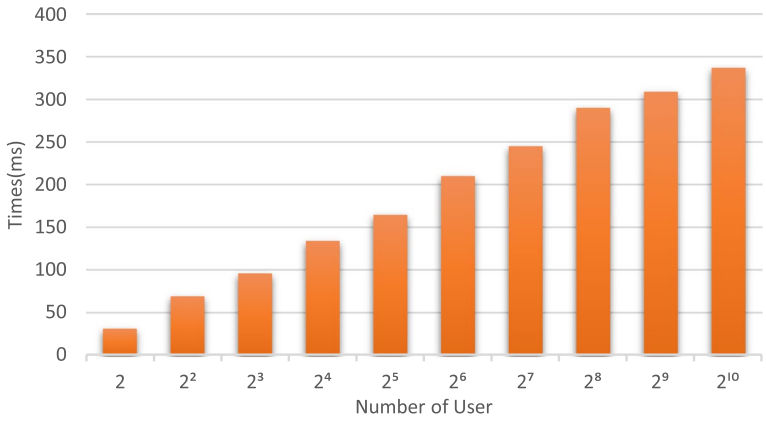

Fig. 5 Time cost of KeyUp algorithm

Conclusion

We propose an RM-CP-ABE scheme suitable for CFS systems. It implements efficient dynamic management of users and supports distributed frameworks. In addition, it is also resistant to decryption key exposure attack. By using the game sequence, we prove the security against CPA attacks and collusion attacks under the random oracle model. We also conducted an implementation to demonstrate the practicability of our RLWE-based RM-CP-ABE scheme. The future works will focus on the construction of secure and efficient CP-ABE scheme under the standard model. At the same time, we can also consider constructing a CP-ABE scheme with more flexible access policies.

Authors' contributions

Boxue Huang: Methodology, Software, Writing - original draft. Juntao Gao: Conceptualization, Supervision, Writing - review & editing. XueLian Li: Formal analysis, Funding acquisition. The author(s) read and approved the final manuscript.

Funding

This work is supported in part by the Key Research and Development Program of Shaanxi (No. 2021ZDLGY06-04), Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201802).

Availability of data and materials

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare no competing interests.

Received: 17 November 2022 Accepted: 25 February 2023
Published online: 11 March 2023

References

1. Backendal M, Haller M, Paterson KG (2023) Mega: malleable encryption goes awry. 44th IEEE Symposium on Security and Privacy (SP 2023). San Francisco
2. Chen E, Zhu Y, Liang K, Yin H (2023) Secure remote cloud file sharing with attribute-based access control and performance optimization. In: IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp 579–594. https://doi.org/10.1109/TCC.2021.3104323
3. Li J, Wang S, Li Y, Wang H, Wang H, Wang H, Chen J, You Z (2019) An efficient attribute-based encryption scheme with policy update and file update in cloud computing. IEEE Trans Ind Inf 15(12):6500–6509
4. Deng H, Qin Z, Wu Q, Guan Z, Yin H (2022) Revocable attribute-based data storage in mobile clouds. In: IEEE Transactions on Services Computing, vol. 15, no. 2, pp 1130–1142. https://doi.org/10.1109/TSC.2020.2984757
5. Xue K, Chen W, Li W, Hong J, Hong P (2018) Combining data owner-side and cloud-side access control for encrypted cloud storage. IEEE Trans Inf Forensic Secur 13(8):2062–2074
6. Wang S, Zhou J, Liu JK, Yu J, Chen J, Xie W (2016) An efficient file hierarchy attribute-based encryption scheme in cloud computing. IEEE Trans Inf Forensic Secur 11(6):1265–1277
7. Joseph D, Misoczki R, Manzano M, Tricot J, Pinuaga FD, Lacombe O, Leichenauer S, Hidary J, Venables P, Hansen R (2022) Transitioning organizations to post-quantum cryptography. Nature 605(7909):237–243
8. Yang Y, Sun J, Liu Z, Qiao Y (2022) Practical revocable and multi-authority cp-abe scheme from rlwe for cloud computing. J Inf Secur Appl 65:103108
9. Wang S, Zhang X, Zhang Y (2018) Efficient revocable and grantable attribute-based encryption from lattices with fine-grained access control. IET Inf Secur 12(2):141–149
10. Xu S, Yang G, Mu Y (2019) Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation. Inf Sci 479:116–134
11. Xu S et al (2023) A secure EMR sharing system with tamper resistance and expressive access control. In: IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 1, pp 53–67. https://doi.org/10.1109/TDSC.2021.3126532
12. Datta P, Komargodski I, Waters B (2021) Decentralized multi-authority ABE for DNFs from LWE. In: Canteaut A, Standaert FX (eds) Advances in Cryptology – EUROCRYPT 2021. EUROCRYPT 2021. Lecture Notes in Computer Science, vol 12696. Springer, Cham. https://doi.org/10.1007/978-3-030-77870-5_7
13. Zhang G, Qin J, Qazi S (2015) Multi-authority attribute-based encryption scheme from lattices[J]. J Univers Comput Sci 21(3):483–501. https://doi.org/10.3217/jucs-021-03-0483

14. Zhu Y, Hu H, Ahn GJ, Huang D, Wang S (2012) Towards temporal access control in cloud computing. Proceedings IEEE INFOCOM, Orlando, pp 2576–2580. https://doi.org/10.1109/INFCOM.2012.6195656
15. Zhu S, Yang X, Wu X (2013) Secure cloud file system with attribute based encryption. 2013 5th International Conference on Intelligent Networking and Collaborative Systems, Xi'an, pp 99–102. https://doi.org/10.1109/INCoS.2013.22
16. Zhu S, Gong G (2014) Fuzzy authorization for cloud storage. IEEE Trans Cloud Comput 2(4):422–435
17. Zhang J, Zhang Z, Ge A (2012) Ciphertext policy attribute-based encryption from lattices. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. ASIACCS '12. Association for Computing Machinery, New York, pp 16–17. https://doi.org/10.1145/2414456.2414464
18. Chen Z, Zhang P, Zhang F, Huang J (2017) Ciphertext policy attribute-based encryption supporting unbounded attribute space from r-lwe. KSII Trans Internet Inf Syst (TIIS) 11(4):2292–2309
19. Gür KD, Polyakov Y, Rohloff K, Ryan GW, Sajjadpour H, Savaş E (2018) Practical applications of improved gaussian sampling for trapdoor lattices. IEEE Trans Comput 68(4):570–584
20. Ibraimi L, Petkovic M, Nikova S, Hartel P, Jonker W (2009) Mediated ciphertext-policy attribute-based encryption and its application. In: Youm HY, Yung M (eds) Information Security Applications. WISA 2009. Lecture Notes in Computer Science, vol 5932. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-10838-9_23
21. Sahai A, Seyalioglu H, Waters B (2012) Dynamic credentials and ciphertext delegation for attribute-based encryption. In: Safavi-Naini R, Canetti R (eds) Advances in Cryptology – CRYPTO 2012. CRYPTO 2012. Lecture Notes in Computer Science, vol 7417. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32009-5_13
22. Xu S, Yang G, Mu Y, Liu X (2019) A secure iot cloud storage system with fine-grained access control and decryption key exposure resistance. Futur Gener Comput Syst 97:284–294
23. Yang K, Wu G, Dong C, Fu X, Li F, Wu T (2020) Attribute based encryption with efficient revocation from lattices. Int J Netw Secur 22(1):161–170
24. Takayasu A, Watanabe Y (2017) Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance. In: Pieprzyk J, Suriadi S (eds) Information Security and Privacy. ACISP 2017. Lecture Notes in Computer Science, vol 10342. Springer, Cham. https://doi.org/10.1007/978-3-319-60055-0_10
25. Dong X, Hu Y, Wang B, Liu M, Gao W (2021) Lattice-based revocable attribute-based encryption with decryption key exposure resistance. IET Inf Secur 15(6):428–441
26. Chase M (2007) Multi-authority attribute based encryption. In: Vadhan SP (ed) Theory of cryptography conference. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 515–534
27. Rouselakis Y, Waters B (2015) Efficient statically-secure large-universe multi-authority attribute-based encryption. International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 315–332
28. Agrawal S, Boneh D, Boyen X (2010) Efficient lattice (H)IBE in the standard model. In: Gilbert H (ed) Advances in Cryptology – EUROCRYPT 2010. EUROCRYPT 2010. Lecture Notes in Computer Science, vol 6110. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13190-5_28
29. Dai W, Doröz Y, Polyakov Y, Rohloff K, Sajjadpour H, Savaş E, Sunar B (2017) Implementation and evaluation of a lattice-based key-policy abe scheme. IEEE Trans Inf Forensic Secur 13(5):1169–1184
30. Naor D, Naor M, Lotspiech J (2001) Revocation and tracing schemes for stateless receivers. In: Kilian J (ed) Annual International Cryptology Conference, vol 2139. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 41–62

Publisher's Note