

# Revocable Multiple Groups Outsourcing Data Integrity Auditing Scheme on Lattices

xueyan Liu (✉ [liuxy@nwnu.edu.cn](mailto:liuxy@nwnu.edu.cn))

Northwest Normal University

xiaoyan Zhang

Northwest Normal University

jia Wang

Northwest Normal University

xiliang Li

Northwest Normal University

---

## Research Article

**Keywords:** Data integrity, Lattices, Multiple groups, Conditional Anonymity, Traceability

**Posted Date:** May 29th, 2023

**DOI:** <https://doi.org/10.21203/rs.3.rs-2981150/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

**Additional Declarations:** No competing interests reported.

---

# Revocable Multiple Groups Outsourcing Data Integrity Auditing Scheme on Lattices

Xueyan Liu, Xiaoyan Zhang, Jia Wang, Xiliang Li

## Abstract

The massive data scale and diverse data types give the data greater value. In the context of the big data era, the realization of outsourced data sharing is conducive to the maximum potential of the data. However, most of the existing outsourcing data sharing schemes only consider data sharing among members in a single group, and ignore the scenario of outsourcing data sharing among multiple groups. In addition, most data integrity auditing schemes use large integer decomposition, discrete logarithms, and other difficult problems. With the rapid development of quantum computing and quantum communication technologies, their security is challenged. In view of these, a lattice-based user revocable multiple groups outsourcing data integrity auditing scheme is proposed. Lattice based delegation technique is used to generate user private keys, which can resist the attack of key exposure, optimize the inherent anonymity and traceability of the group signature. We also combine the technique with identity-based encryption to protect user identity and avoid complex certificate management issues. The use of rejection sampling algorithm in the signature phase to improve signature and verification efficiency. Furthermore, three cases of batch auditing are discussed according to the source of users. At last, the LPR encryption algorithm is adopted to ensure that the group manager can open the group signature by tracing keys and thus enable identity tracing. When a user is revoked, the manager uses the SamplePre algorithm to regenerate partial private keys for the non-revoked users and ensure the security and confidentiality of data. The scheme satisfies conditional anonymity, unforgeability and traceability on the basis of assumptions of RSIS and RLWE problems.

**Keywords** Data integrity, Lattices, Multiple groups, Conditional Anonymity, Traceability

## 1 Introduction

Cloud storage meets the needs of enterprises and individuals for data use, storage, sharing, and protection. These features can achieve sharing of outsourced data and maximize the value of data. Uploading data to the cloud server can reduce the burden of local storage. However, it can also pose some problems due to the loss of physical control of the data by the user. For example, whether the data is secure, whether there is a privacy breach, and whether the data can be recovered if the cloud server fails. Therefore, it is very important to ensure outsourced data integrity.

Publicly verified cloud storage schemes allow outsourced data integrity auditing to be delegated to trusted third-party auditors (TPA), which is preferable. Many schemes are proposed to verify the integrity of outsourced data [1-7]. However, some existing schemes present the following weaknesses: First, there are certificate management problems based on the public key infrastructure (PKI) [8]. Second, most data integrity verification schemes are based on traditional cryptographic hard problems that are no longer secure with the arrival of quantum computers [9]. Finally, only outsourcing the practical scenarios of data outsourcing shared by multiple groups [10]. There are numerous application scenarios of

multiple groups outsourcing data sharing. For example, in telemedicine, several medical institutions need to establish shared medical databases in order to improve medical care, so that they can obtain more reliable and comprehensive data to make accurate treatments for patients. In order to realize literature sharing, several universities jointly establish mobile libraries and upload their literature to the mobile libraries. So that authorized users can get the relevant information they need anytime and anywhere. In above scenarios, users of different groups share data through cloud storage.

Data confidentiality in the telemedicine process ensures that sensitive information of patients is legally accessed, impersonation of patient identities is avoided, and data integrity can help health-care personals make important decisions. Therefore, the sharing outsourced data across multiple groups must consider the following issues: (1) To ensure the integrity, authenticity and validity of the uploaded data, the user can delegate TPA to audit the data in the cloud server. (2) Users who leave the group cannot share data and send data auditing requests because the group manager revokes their identities. Note that multiple groups work together to share outsourced data. Managers of these groups enjoy equal rights and the user's revocation list needs to be broadcast among the groups. (3) Users are completely anonymous during data sharing and auditing to protect user privacy. The manager of the group can track the real identity of the user to achieve traceability when a dispute occurs.

### A. Related Work

The provable data possession (PDP) mechanism, firstly proposed by Ateniese [11]. Users or TPA can verify the integrity of data without downloading the original data, which greatly reduces the computational and communication overhead of auditing. PKI-based public auditing schemes have complex key management and certificate validation problems.

Xueyan Liu is with the school of Computer Science and Engineering, Northwest Normal University, Lanzhou, China (e-mail: liuxy@nwnu.edu.cn).

Xiaoyan Zhang is currently pursuing the master's degree with the school of mathematics and statistics, Northwest Normal University, Lanzhou, China (e-mail: zxiaoyan97@163.com).

Jia Wang is currently pursuing the master's degree with the school of Computer Science and Engineering, Northwest Normal University, Lanzhou, China (e-mail: 1907301246@qq.com).

Xiliang Li is currently pursuing the master's degree with the school of Computer Science and Engineering, Northwest Normal University, Lanzhou, China (e-mail: 335901582@qq.com).

To reduce the requirements for certificates in PKI schemes and to ensure that curious TPA cannot access user information from cloud servers, many identity-based data integrity auditing schemes have been proposed [12-14]. Shang et al. [16] proposed a data integrity auditing scheme supporting dynamic operations, in which Merkle Hash Tree was used to data modification, insertion and deletion data blocks. In this scheme, the TPA is honest and trustworthy, and therefore, it is difficult to resist attacks by untrustworthy third parties. Garg et al. [17] proposed a cloud computing auditing protocol that is publicly verified and supports dynamic manipulation of data. The protocol minimizes the computational complexity on the user side during the auditing phase and guarantees the efficiency of the proposed scheme. In above identity-based schemes the user's public key is related to its identity, simplifying the certificate management problem in PKI. However, the above schemes only discuss the dynamic operation of data without considering the revocability of user, and ignores the application scenarios of data sharing among multiple groups.

To solve the data integrity auditing of group users in cloud storage, Wang et al. [18] proposed the first identity privacy-preserving public auditing scheme for group data. In the scheme ring signature is used to protect user's identity privacy, realize batch audit and improve the efficiency of data block verification. However, computation overhead of TPA is linearly related to the number of group users, and it is difficult to be applied to outsourced data integrity auditing of large groups. Tian et al. [19] proposed public auditing of shared data through efficient and secure group management, which realizes batch auditing and reduces computational costs. The scheme protects data privacy, supports user identity tracking and enables batch auditing. Shen et al. [20] proposed a new traceable group data sharing scheme. The scheme supports multiple users using key exchange protocols and group signatures. This scheme supports dynamic changes of users in many-to-many data sharing mode and is suitable for group data sharing in cloud environments.

All above schemes are based on the discrete logarithm problem and cannot resist quantum attacks. In recent years, with the development of quantum computers, group signatures constructed based on traditional number theory problem are no longer secure. Therefore, establishing lattice-based cryptography with worst-case hardware assumptions has become a hot research topic. Xie et al. [10] proposed a lattice-based dynamic group signature auditing scheme in IoT, which allows users to dynamically join the group while supporting member revocation. The group manager revokes group members by updating the group public key. However, this scheme only considers a single group and does not consider the data auditing among multiple groups.

Therefore, a lattice-based user revocable data integrity auditing scheme for multiple groups outsourcing is constructed to address the limitations of existing schemes and meet the needs of actual applications and problems of data reliability when information comes from various groups and users. The TPA will perform data auditing in the following cases: (1) Different files of the same user. This auditing process is similar to the existing auditing scheme; (2) Files from different users in the same group. In this case, members

of the same group generate valid signatures for the data files, and the TPA cannot know exactly which user generated the signature, that is, the curious TPA cannot access any privacy information about the user; (3) Files from different users in different groups. Outsourced data can be shared among different groups, so members from different groups can initiate audit requests.

## B. Contributions

We combine lattice-based group signature with identity-based encryption to propose a revocable multiple groups outsourcing data integrity auditing scheme on lattices. The main contributions are as follows:

(1) A data integrity auditing scheme of user revocable multiple groups outsourcing based on lattice cryptography is put forward. Identity-based data integrity auditing solves the public key certificate management problem. Lattice based delegation technology is adopted to generate user private keys and resist the attack of key exposure.

(2) The signature phase uses a rejection sampling algorithm to generate signatures to improve the computational efficiency of the signing and verification process. User conditional anonymity is achieved by using group signatures and identity tracking is achieved by using LPR encryption algorithm [21] to ensure that group managers can open group signatures by tracing the key.

(3) We can aggregate multiple verification equations about different users into a single equation to efficiently process multiple data integrity audit requests.

## 2 Preliminaries

### A. Lattices

*Definition 1:* A set of linearly independent column vectors  $B = \{b_1, b_2, \dots, b_n\} \in \mathbb{R}^{m \times m}$  generates an  $m$ -dimensional full-rank lattice  $\mathcal{A}$ , which is defined as

$$\mathcal{A} = \{Bc = \sum_{1 \leq i \leq m} c_i b_i, c_i \in \mathbb{Z}\}.$$

Here  $B = \{b_1, b_2, \dots, b_n\}$  is a basis of the lattice  $\mathcal{A}$ , the length of the basis  $B$  is the length of the longest vector in  $B$ , denote as  $\|B\| = \max_{1 \leq i \leq m} \|b_i\|$ .

*Definition 2:* For prime  $q$ , matrix  $A \in \mathbb{Z}_q^{n \times m}$ , the definition of the  $q$ -modular integer lattices are defined as follows.

- 1)  $\mathcal{A}_q(A) = \{x \in \mathbb{Z}^m : \exists c \in \mathbb{Z}_q^n, x = A^T c \bmod q\}.$
- 2)  $\mathcal{A}_q^\perp(A) = \{e \in \mathbb{Z}^m : Ae = 0 \bmod q\}.$
- 3)  $\mathcal{A}_q^\gamma(A) = \{e \in \mathbb{Z}^m : Ae = y \bmod q\}.$

### B. Discrete Gaussian distribution

*Definition 3:*  $s \in \mathbb{R}^m$  is standard deviation, vector  $c \in \mathbb{Z}^m$  is center, the Gaussian function is defined as

$$g_{s,c}(x) = e^{-\frac{\pi^2 \|x-c\|^2}{2s^2}}.$$

The discrete Gaussian distribution over  $\mathcal{A}$  with center  $c$  and parameters  $s$  is defined as

$$G_{\mathcal{A},s,c}(x) = \frac{g_{s,c}(x)}{\sum_{x \in \mathcal{A}} g_{s,c}(x)}.$$

### C. Some algorithms on lattice

**Lemma 1:** For any prime  $q \geq 2$ ,  $m \geq \lceil 5n \log q \rceil$ , there exists a probabilistic polynomial-time (PPT) algorithm *TrapGen* ( $q, n, m$ ) that returns  $A \in \mathbb{Z}_q^{n \times m}$  and  $T \in \mathbb{Z}_q^{m \times m}$ , where  $A$  is statistically close to a uniform matrix in  $\mathbb{Z}_q^{n \times m}$  and  $T$  is a basis for lattice  $\Lambda_q^\perp(A)$  with  $\|T\| \leq O(n \log q)$  and  $\|T\| \leq O(\sqrt{n \log q})$ .

**Lemma 2:** Let  $m \geq n$  be an integer and  $q$  be prime. Let matrix  $A \in \mathbb{Z}_q^{m \times n}$  define a lattice  $\Lambda^\perp(A)$  and  $B$  be a basis of  $\Lambda^\perp(A)$ . If  $s \geq \|\tilde{B}\| \cdot \omega(\sqrt{\log n})$ , then for any  $u \in \mathbb{Z}_q^n$ , there is a PPT algorithm *SamplePre* ( $A, B, s, u$ ) that outputs a vector  $v \in \Lambda^u(A)$  from a distribution that is statistical close to uniform  $G_{A,s,c}(x)$ .

**Lemma 3:** On input a matrix  $A \in \mathbb{Z}_q^{n \times m}$ , a short basis  $T_A \in \mathbb{Z}_q^{m \times m}$  of lattice  $\Lambda_q^\perp(A)$  and an invertible matrix  $R \in \mathbb{Z}_q^{m \times m}$  sampled from the distribution  $D_{m \times m}$  which is defined as  $(D_{Z^m, \sigma_R})^m$  and a Gaussian parameter  $\sigma \geq \|\tilde{T}_A\| \sigma_R \sqrt{m} \omega(\sqrt{\log^3 m})$ , there exists a PPT algorithm *NewBasisDel* ( $A, T_A, R, \sigma$ ), outputs a randomized lattice basis  $T$  of  $\Lambda_q^\perp(AR^{-1})$  [22].

**Lemma 4: Rejection Sampling Technique** [23].

Input: Message  $u$ , randomly sample a matrix  $A$  from  $\mathbb{Z}_q^{n \times m}$ ,  $S$  sampled from  $\{-d, \dots, 0, \dots, d\}^{m \times k}$ , where  $d \ll q^{n/m}$ ,  $k \ll m \in \mathbb{Z}$ ,  $H: \{0,1\}^* \rightarrow \{v: v \in \{-1,0,1\}^k, \|v\| < \kappa\}$ , where  $\kappa$  is constant and  $2^\kappa$ . Then there exists a constant  $M = O(1)$ .

Output: Vector  $z$  and  $c$

1. Obtain  $y$  randomly from  $D_\sigma^m$
2.  $c = H(Ay, u)$
3.  $z = Sc + y$
4. return  $(z, c)$  with probability  $\min\left(\frac{D_\sigma^m(z)}{M D_{Sc, \sigma}(z)}, 1\right)$ .

### D. Ring Small Integer Solution Problem [24, 25]

**Definition 4:** Given a security parameter  $n$  and a uniform random matrix  $A = [a_1, a_2, \dots, a_m] \in \mathbb{R}_q^{1 \times m}$ , find a nonzero vector  $u = (u_1, u_2, \dots, u_m)^T \in \mathbb{R}^m$  satisfying  $\|u\|_\infty \leq \beta$  and  $Au = \sum a_i u_i = 0 \bmod q$ . The *RSIS* problem is at least as difficult as the *Ideal-SIS* problem when the parameters satisfy  $q \geq \gamma \sqrt{n}/a \log n$ ,  $\beta = \text{plog}(n)$ ,  $m > \log q / \log(2\beta)$ ,  $\gamma = 16\beta m n \log^2 n$ .

### E. Decision Ring Learn With Error problem [21]

**Definition 5:** Given  $n, m \geq 1$ ,  $q \geq 2$ , the probability distribution  $\chi$  over  $\mathbb{R}$ . For  $s \in \mathbb{R}_q$ ,  $a \leftarrow \mathbb{R}_q$  and  $e \leftarrow \chi$  are chosen randomly and the distribution generated by  $b = as + e$  is  $A_{s,\chi}$ . The *DRLWE* problem is to discriminate whether the  $m$  samples are taken from the distribution  $A_{s,\chi}$  or from the distribution  $U(\mathbb{R}_q \times \mathbb{R}_q)$ .

### F. System model

The system model includes five entities: user, GMs (group managers), CSP (the cloud service provider), KGC (the key generation center) and TPA (the third-party auditor).

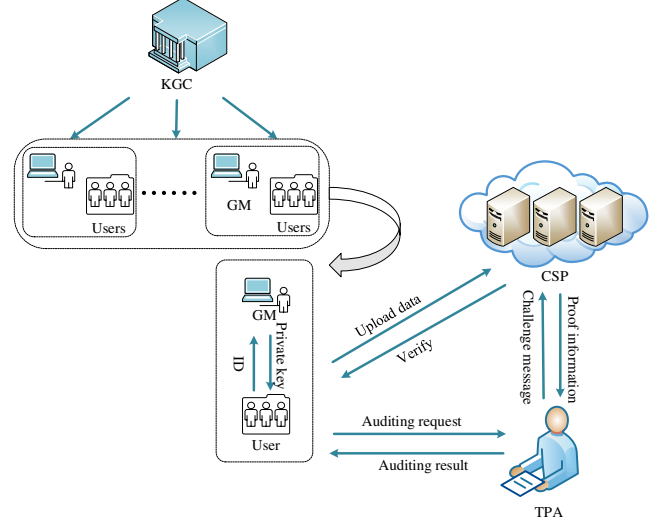


**Fig. 1** The system model

(1) User: Users are group members and have a lot of data that needs to be stored.

(2) GMs: Each group has a manager who generates public-private key pairs for group members; when a user leaves the group, the group manager is responsible for revoking the user.

(3) CSP: CSP has powerful storage capabilities and is responsible for storing and maintaining user data.

(4) KGC: It is a trusted entity that generates the group public key, group private key and traceability key for each group manager.

(5) TPA: TPA is responsible for completing the integrity verification of the data and providing feedback to the user.

## 3 Construction

### A. Our Scheme.

In our scheme, given polynomial rings  $\mathbb{R} = \mathbb{Z}[x]/(x^n + 1)$  and  $\mathbb{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ . Select the boundary parameter  $\xi = \tilde{O}(n)$  and define the error distribution  $\chi$  as the distribution on  $\mathbb{R}$  with  $\xi$  as the boundary. The maximum number of group members for each group is  $N$ . Set the public parameters as below. Gaussian parameter  $\sigma = \text{poly}(n)$ , the infinity norm bound for signature is  $\beta = \text{poly}(n)$ , prime  $q, m$  and  $l$ . The specific structure of our scheme is as follows.

### Key generation stage

This stage includes two algorithms: Setup and KeyGen, the process is shown in Fig.2.

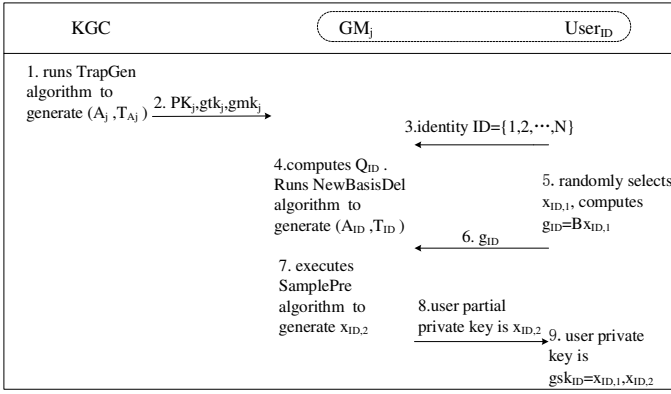


**Fig. 2** Process of key generation

**Setup:** On input the security parameter  $n$ , the KGC runs the following algorithm to generate the group public key, traceability key and group private key for each  $GM_j$  which denotes the manager of the  $j$ -th group,  $j \in [1, K]$ , where  $K$  is the maximum number of groups.

- The KGC generates matrixes  $A_j \in \mathbb{R}_q^{1 \times m}$  and short basis  $T_{A_j} \in \mathbb{Z}_q^{m \times m}$  of  $A_q^\perp(A_j)$  by running algorithm  $TrapGen(n, m, q)$ .
- The KGC randomly selects  $a \leftarrow \mathbb{R}_q^{1 \times l}$ ,  $e \leftarrow \chi^l$ ,  $v_j \in \chi$ ,  $u \leftarrow \mathbb{R}_q$ ,  $B \leftarrow \mathbb{R}_q^{1 \times m}$  and computes  $b_j = (av_j + e) \bmod q$ .
- Define five secure cryptographic hash functions:  $H_1: \{0,1\}^* \rightarrow \mathbb{R}_q^{m \times m}$ ,  $H_2: \mathbb{R}_q^{1 \times m} \times \mathbb{R}_q^m \rightarrow \mathbb{R}_q$ ,  $H_3: \{0,1\}^* \times \{0,1\}^l \times [1, K] \rightarrow \mathbb{R}_q$ ,  $H_4: \mathbb{R}_q \times \mathbb{R}_q \rightarrow \mathbb{R}_q$ ,  $H_5: \mathbb{R}_q \times \mathbb{R}_q \times \mathbb{R}_q^l \rightarrow \{-1, 0, 1\}^m$ .
- Finally, the KGC publishes the group public keys  $PK_j = (A_j, B, a, b_j, u)$ , group traceability key  $g_{tk_j} = v_j$  and the group managers' secret key  $gmk_j = T_{A_j}$ .

**KeyGen:** Enter the group public key  $PK_j$ , the master key  $T_{A_j}$  and the user identity  $ID$ , the  $GM_j$  and the user to execute the following algorithm.

- According to user identity  $ID = \{1, 2, \dots, N\} \in \{0,1\}^*$ , the  $GM_j$  computes  $Q_{ID} = H_1(ID || j)$ , where  $j \in [1, K]$  and runs  $NewBasisDel(A_j, T_{A_j}, Q_{ID}, \sigma)$  algorithm to obtain a random short basis  $T_{ID}$  of lattice  $A_q^\perp(A_{ID})$ . Here  $A_{ID} = A_j Q_{ID}^{-1} \in \mathbb{R}_q$ .
- The user randomly selects  $x_{ID,1} \leftarrow D_\sigma^m$ , computes  $g_{ID} = Bx_{ID,1} \bmod q \in \mathbb{R}$  and sends  $g_{ID}$  to the  $GM_j$ .
- The  $GM_j$  compares  $g_{ID}$  with the local list  $\{g_i\}_{i=1}^N$ . If the value of  $g_{ID}$  is equal to the known value of  $g_i$ , the user performs step (b) again. Otherwise, the  $GM_j$  will store  $g_{ID}$  in the list  $\{g_i\}_{i=1}^N$ .
- The  $GM_j$  executes  $SamplePre(A_{ID}, T_{ID}, u - g_{ID}, \sigma)$  algorithm to obtain user partial private key  $x_{ID,2}$ .
- The  $GM_j$  publishes revocation list  $RL_j$ , initially empty. The user private key is  $gsk_{ID} = (x_{ID,1}, x_{ID,2})$ .

### Data submission stage

At this stage, users will outsource their encrypted and signed data to CSP, as shown in Fig. 3 for the detailed process.

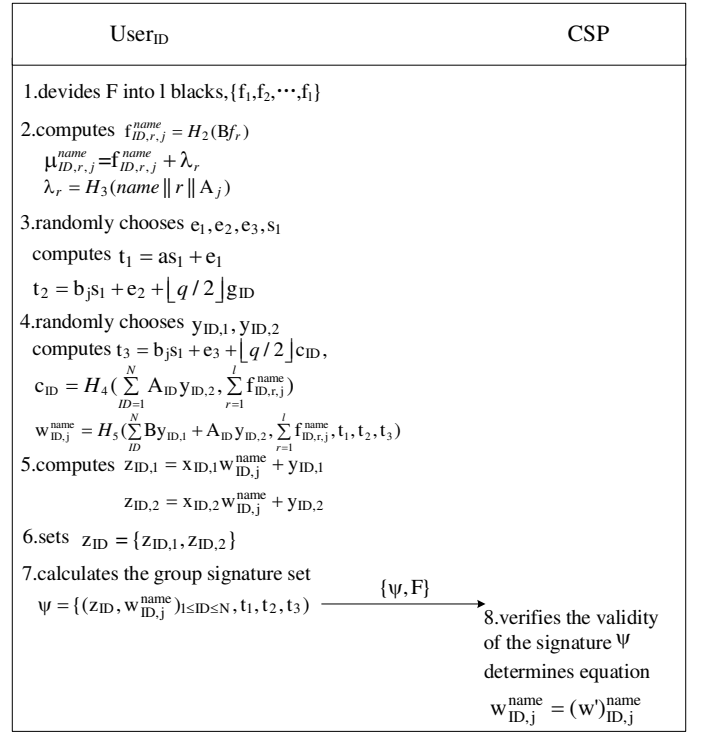


**Fig. 3** Process of data submission

**SignGen:** Enter the group public key  $PK_j$ , the user private key  $gsk_{ID}$ , and a data file  $F$ , with name  $name \in \{0,1\}^*$  the user to execute the following algorithm.

- The user divides  $F$  into  $l$  blocks and marked as  $\{f_1, f_2, \dots, f_l\}$ , where  $f_r \in \mathbb{R}_q^m$ ,  $r \in [1, l]$ , computes  $f_{ID,r,j}^{name} = H_2(Bf_r)$ ,  $\lambda_r = H_3(name || r || A_j) \in \mathbb{R}_q$  and  $\mu_{ID,r,j}^{name} = f_{ID,r,j}^{name} + \lambda_r$ .
- The user randomly chooses  $e_1, e_2, e_3 \leftarrow \chi^l$ ,  $s_1 \leftarrow \chi$  and computes  $t_1 = as_1 + e_1$ ,  $t_2 = b_j s_1 + e_2 + \lfloor q/2 \rfloor g_{ID}$ .
- The user randomly chooses  $y_{ID,1}, y_{ID,2} \leftarrow D_\sigma^m$ , computes the following equations:  
 $t_3 = b_j s_1 + e_3 + \lfloor q/2 \rfloor c_{ID}$ ,  $c_{ID} = H_4(\sum_{ID=1}^N A_{ID} y_{ID,2}, \sum_{r=1}^l f_{ID,r,j}^{name})$ ,  
 $w_{ID,j}^{name} = H_5(\sum_{ID=1}^N B y_{ID,1} + A_{ID} y_{ID,2}, \sum_{r=1}^l f_{ID,r,j}^{name}, t_1, t_2, t_3)$ .
- The user uses the rejection sampling algorithm to get  $z_{ID,1} = x_{ID,1} w_{ID,j}^{name} + y_{ID,1}$  and  $z_{ID,2} = x_{ID,2} w_{ID,j}^{name} + y_{ID,2}$ , let  $z_{ID} = (z_{ID,1}, z_{ID,2})$ .
- Output the group signature  $\Psi = \{(z_{ID}, w_{ID,j}^{name})_{1 \leq ID \leq N}, t_1, t_2, t_3\}$  of the user on the file  $F$ .
- Then, the user sends  $\{\Psi, F\}$  to the CSP.
- The CSP verifies the validity of the signature by computing  $(w')_{ID,j}^{name} = H_5(\sum_{ID=1}^N (Bz_{ID,1} + A_{ID} z_{ID,2} - u w_{ID,j}^{name}), \sum_{r=1}^l f_{ID,r,j}^{name}, t_1, t_2, t_3)$  to determine whether equation  $w_{ID,j}^{name} = (w')_{ID,j}^{name}$  holds. If the verification is passed,  $\{\Psi, F\}$  is stored into the CSP; otherwise, the CSP refuses to store it.

### Data Auditing Stage

**GenChal:** Each user in groups that participates in data sharing can request the TPA to verify the integrity of the

shared data. The TPA receives the auditing request and generates the challenge message and sends it to the CSP.

- (a) The TPA picks a random  $c$ -element subset  $I$  of set  $I \subseteq [1, l]$ . For each element  $i \in I$ , the TPA chooses a random value  $h_i = \{h_1, h_2, \dots, h_c\} \in \{0, 1\}^c$ .
- (b) The TPA sends  $chal = (i, \{h_i\}_{i \in I})$  to the CSP.

In our scheme, since outsourced data is shared among multiple groups, the data auditing can be divided into the following three cases: 1) Data files from the same user. 2) Data files from different users in the same group. 3) Data files from different users in different groups, Fig. 4 illustrates three audit processes. In these three cases, the following descriptions are given according to the different validation needs:

$$\begin{aligned}\tilde{F}_1 &= \sum_{name=1}^m \sum_{r=1}^c h_r f_{ID,r,j}^{name}, \\ \tilde{F}_2 &= \sum_{ID=1}^N \sum_{r=1}^c h_r f_{ID,r,j}^{name}, \\ \tilde{F}_3 &= \sum_{j=1}^K \sum_{ID=1}^N \sum_{r=1}^c h_r f_{ID,r,j}^{name}.\end{aligned}$$

The following will describe three different cases:

1) Data files from the same user

**ProofGen**: After the CSP receives the challenge message, it calculates (1), (2), and sends  $proof = \{(f_{ID,r,j}^{name}, z_{ID}, W), \tilde{F}_1\}$  to the TPA.

$$\tilde{F}_1 = \sum_{name=1}^m \sum_{r=1}^c h_r f_{ID,r,j}^{name}$$

$$(1) \quad W = \sum_{name=1}^m (w')_{ID,j}^{name}$$

**ProofVerify**: Computes  $\sum_{r=1}^c h_r \lambda_r = \lambda$ , the TPA runs the following algorithm:

- (a) The TPA checks whether the following inequation  $\|z_{ID,1}\| \leq 2\sigma\sqrt{n}$  and  $\|z_{ID,2}\| \leq 2\sigma\sqrt{n}$  holds. If “True”, continues to perform the following steps. Otherwise, stop.
- (b) Checks whether the following equation (3) holds. If “True”, continues. Otherwise, stop.

$$W = \sum_{name=1}^m w_{ID,j}^{name} = \sum_{name=1}^m (w')_{ID,j}^{name}$$

(3)

- (c) Validates whether the following equation (4) holds. If the verification is valid then return “1”, otherwise the verification fails back to “0”.

$$\sum_{name=1}^m \sum_{r=1}^c h_r \mu_{ID,r,j}^{name} \bmod q = \tilde{F}_1 + \lambda$$

(4)

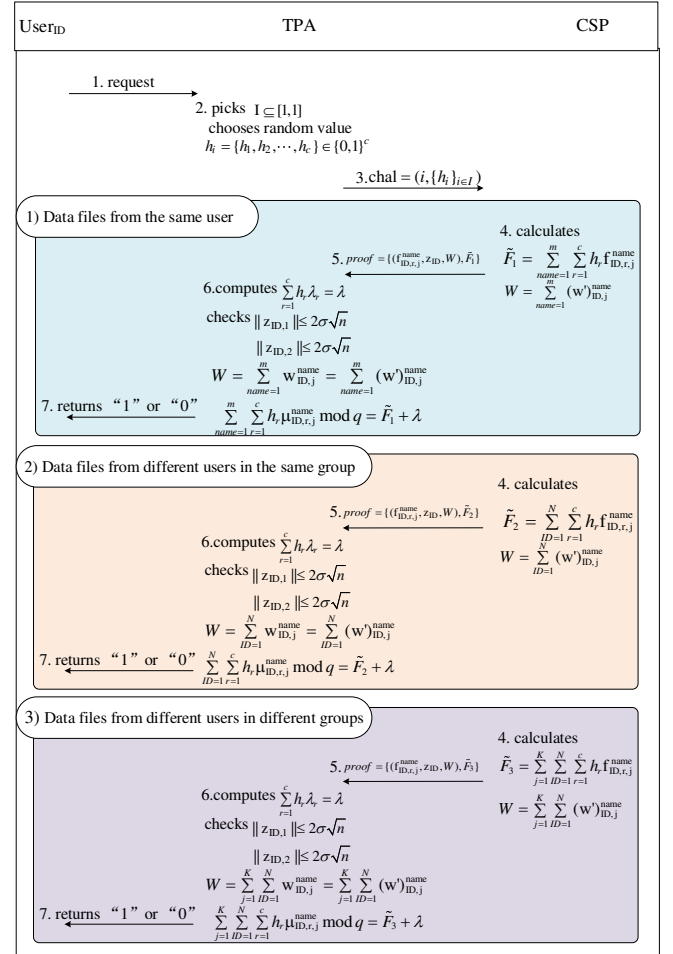


Fig. 4 Process of data auditing

2) Data files from different users in the same group

**ProofGen**: After the CSP receives the challenge message, it calculates (5), (6), and sends  $proof = \{(f_{ID,r,j}^{name}, z_{ID}, W), \tilde{F}_2\}$  to the TPA.

$$\tilde{F}_2 = \sum_{ID=1}^N \sum_{r=1}^c h_r f_{ID,r,j}^{name}$$

$$(5) \quad W = \sum_{ID=1}^N (w')_{ID,j}^{name}$$

(6)

**ProofVerify**: The TPA computes  $\sum_{r=1}^c h_r \lambda_r = \lambda$  and runs the following algorithm:

- (a) The TPA checks whether the following inequation  $\|z_{ID,1}\| \leq 2\sigma\sqrt{n}$  and  $\|z_{ID,2}\| \leq 2\sigma\sqrt{n}$  holds. If “True”, continues to perform the following steps. Otherwise, stop.
- (b) Checks whether the following equation (7) holds. If “True”, continues. Otherwise, stop.

$$W = \sum_{ID=1}^N w_{ID,j}^{name} = \sum_{ID=1}^N (w')_{ID,j}^{name}$$

(7)

- (a) Validates whether the following equation (8) holds. If the verification is valid then return “1”, otherwise the verification fails back to “0”.

$$\sum_{ID=1}^N \sum_{r=1}^c h_r \mu_{ID,r,j}^{name} \bmod q = \tilde{F}_2 + \lambda$$

(8)

3) Data files from different users in different groups

**ProofGen**: After the CSP receives the challenge message, it calculates (9), (10), sends  $proof = \{(f_{ID,r,j}^{name}, z_{ID}, W), \tilde{F}_3\}$  to the TPA.

$$\tilde{F}_3 = \sum_{j=1}^K \sum_{ID=1}^N \sum_{r=1}^c h_r f_{ID,r,j}^{name} \quad (9)$$

$$W = \sum_{j=1}^K \sum_{ID=1}^N (w')_{ID,j}^{name} \quad (10)$$

**ProofVerify**: Computes  $\sum_{r=1}^c h_r \lambda_r = \lambda$ , the TPA runs the following algorithm:

- The TPA checks whether the following inequation  $\|z_{ID,1}\| \leq 2\sigma\sqrt{n}$ ,  $\|z_{ID,2}\| \leq 2\sigma\sqrt{n}$  holds. If “True”, continues to perform the following steps. Otherwise, stop.
- Checks whether the following equation (11) holds. If “True”, continues. Otherwise, stop.
- Validates whether the following equation (12) holds. If the verification is valid then return “1”, otherwise the verification fails back to “0”.

$$\sum_{j=1}^K \sum_{ID=1}^N \sum_{r=1}^c h_r \mu_{ID,r,j}^{name} \bmod q = \tilde{F}_3 + \lambda \quad (12)$$

### Open and revoke stage

This stage includes two algorithms: Open and Revoke, the process is shown in Fig. 5.

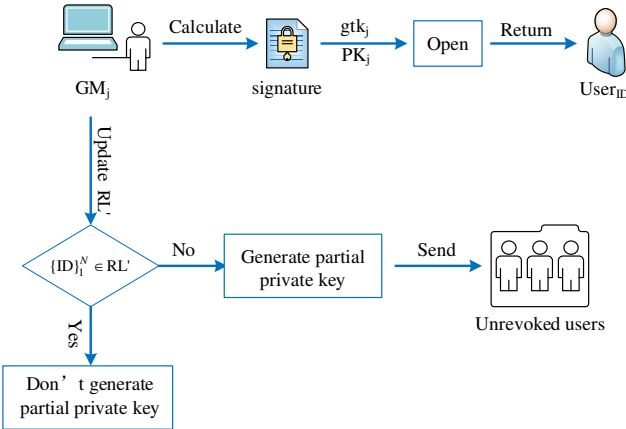


**Fig. 5** Process of open and user revoke

**Open**: On input group public keys  $PK_j$ , signature  $\psi$ , data file  $F$ , group traceability key  $gtk_j$ , identity set of group members  $\{ID\}_1^N$  and revocation list  $RL_j$ , group manager verify valid group signatures in batches:

- The GM calculates  $g_{ID} = t_2 - t_1 v_j$  and  $c_{ID} = t_3 - t_1 v_j$ .
- Calculate the corresponding  $A_{ID}$  according to the user's identity, verify that  $g_{ID}$  satisfies the equation  $c_{ID} = H_4((\sum_{ID=1}^N A_{ID} z_{ID,2} - uw_{ID,j}^{name} + g_{ID} w_{ID,j}^{name}), \sum_{r=1}^c f_{ID,r,j}^{name})$ .

- If the above equation is established, it proves that the opening is successful and the user identity  $ID$  is returned; otherwise, the opening fails and returns  $\perp$ .

**Revoke**: The  $GM_j$  and unrevoked group members perform the following algorithm:

- When a group member is revoked, the  $GM_j$  updates

the user revocation list to  $RL'$ . Unrevoked users randomly selected  $\rho_{ID,1} \leftarrow D_\sigma^m$ , computes  $g'_{ID} = B\rho_{ID,1} \bmod q$  and sends  $g'_{ID}$  to the  $GM_j$ .

- The  $GM_j$  runs  $\text{SamplePre}(A_{ID}, T_{ID}, u - g'_{ID}, \sigma)$  algorithm to generate the unrevoked user partial privacy key  $\rho_{ID,2}$ .
- The new private key of the group member is  $gsk'_{ID} = (\rho_{ID,1}, \rho_{ID,2})$ .

### B. Correctness verification.

$$\begin{aligned} & \sum_{ID=1}^N (Bz_{ID,1} + A_{ID} z_{ID,2} - uw_{ID,j}^{name}) \\ &= \sum_{ID=1}^N (B(x_{ID,1} w_{ID,j}^{name} + y_{ID,1}) + A_{ID}(x_{ID,2} w_{ID,j}^{name} + y_{ID,2}) - uw_{ID,j}^{name}) \\ &= \sum_{ID=1}^N (Bx_{ID,1} w_{ID,j}^{name} + By_{ID,1} + A_{ID} x_{ID,2} w_{ID,j}^{name} + A_{ID} y_{ID,2} - uw_{ID,j}^{name}) \\ &= \sum_{ID=1}^N (By_{ID,1} + A_{ID} y_{ID,2}) + (u - g_{ID}) w_{ID,j}^{name} + g_{ID} w_{ID,j}^{name} - uw_{ID,j}^{name} = \sum_{ID=1}^N (By_{ID,1} + A_{ID} y_{ID,2}). \end{aligned}$$

Therefore, the verification equation  $w_{ID,j}^{name} = (w')_{ID,j}^{name}$  holds. Thus, get the three equations (3), (7), (11) are holds respectively.

From rejection sampling: select  $y_{ID,1}, y_{ID,2} \leftarrow D_\sigma^m$ , thus  $z_{ID,1} = x_{ID,1} w_{ID,j}^{name} + y_{ID,1}$  and  $z_{ID,2} = x_{ID,2} w_{ID,j}^{name} + y_{ID,2}$ . According to Lemma 4, it can be known that  $z_{ID,1}$ ,  $z_{ID,2}$  and Gaussian distribution  $D_\sigma^m$  is statistically indistinguishable, the statistical distance is  $2^{-\omega \log n} / M$ . Thus  $\|z_{ID,1}\| \leq 2\sigma\sqrt{n}$  and  $\|z_{ID,2}\| \leq 2\sigma\sqrt{n}$  holds with overwhelming probability.

$$\begin{aligned} & \sum_{name=1}^m \sum_{r=1}^c h_r \mu_{ID,r,j}^{name} \bmod q \\ &= \sum_{name=1}^m \sum_{r=1}^c (h_r f_{ID,r,j}^{name} + h_r \lambda_r) \\ &= \tilde{F}_1 + \lambda. \end{aligned} \quad (4)$$

$$\begin{aligned} & \sum_{ID=1}^N \sum_{r=1}^c h_r \mu_{ID,r,j}^{name} \bmod q \\ &= \sum_{ID=1}^N (\sum_{r=1}^c h_r f_{ID,r,j}^{name} + h_r \lambda_r) \bmod q \\ &= \tilde{F}_2 + \lambda. \end{aligned} \quad (8)$$

$$\sum_{j=1}^K \sum_{ID=1}^N \sum_{r=1}^c h_r f_{ID,r,j}^{name} \bmod q = \tilde{F}_3 + \lambda. \quad (12)$$

That is, the following three equations (4), (8), (12) hold.

### C. Opening correctness.

Assume that  $\psi = \{(z_{ID}, w_{ID,j}^{name})\}_{1 \leq ID \leq l, t_1, t_2, t_3}$  is a valid signature of user  $ID$  for file  $F$ .

$$\begin{aligned} & \text{Notes, } t_2 - t_1 v_j \\ &= b_j s_1 + e_2 + [q/2] g_{ID} - (a s_1 + e_1) v_j \\ &= e s_1 + e_2 - e_1 v_j + [q/2] g_{ID} \bmod q. \end{aligned}$$

For our scheme,  $\|e\|_\infty, \|e_2\|_\infty, \|e_3\|_\infty, \|s_1\|_\infty \leq \xi$ , the  $\|e s_1 + e_2 - e_1 v_j\|_\infty$  is much smaller than  $[q/2]$ , so the bits of  $g_{ID}$  can be recovered by rounding each coefficient back to either 0 or  $[q/2]$ , whichever is closest ( $\bmod q$ ). Similarly, it is possible to recover the  $c_{ID}$ .

If the signature is a valid signature of a legitimate group member, it can be concluded that  $g_{ID}$ , and the following equation can be satisfied:

$$\begin{aligned}
& \sum_{ID=1}^N (A_{ID} z_{ID,2} - u w_{ID,j}^{name} + g_{ID} w_{ID,j}^{name}) \\
&= \sum_{ID=1}^N (A_{ID} (x_{ID,2} w_{ID,j}^{name} + y_{i,2}) - u w_{ID,j}^{name} + g_{ID} w_{ID,j}^{name}) \\
&= \sum_{ID=1}^N A_{ID} y_{ID,2} + (u - g_{ID}) w_{ID,j}^{name} - u w_{ID,j}^{name} + g_{ID} w_{ID,j}^{name} \\
&= \sum_{ID=1}^N A_{ID} y_{ID,2}.
\end{aligned}$$

Thereby, the equation (13) holds.

## 4 Security analysis

*Theorem 1* (Unforgeability). Our scheme satisfies unforgeability under the random oracle model, if the RSIS problem is hard.

*Proof.* Suppose the adversary  $\mathcal{A}$  forges a valid signature with a non-negligible probability  $\varepsilon$ , then the challenger  $\mathcal{C}$  uses  $\mathcal{A}$  forged signatures to construct non-zero solutions to a set of RSIS problems. First, the challenger  $\mathcal{C}$  enters the security parameters  $n$  and establish  $GMs$  identity set  $\{M_j\}_{j=1}^K$ . The challenger  $\mathcal{C}$  honestly runs the algorithm  $TrapGen(m, q)$  and generates  $gmk_j, gsk_j, gtk_j, pp$ . The challenger  $\mathcal{C}$  sends  $pp, gtk$  to the adversary  $\mathcal{A}$ . The adversary  $\mathcal{A}$  interacts with the challenger  $\mathcal{C}$  to perform hash queries, private key query and signature query.

*Hash query.* The adversary  $\mathcal{A}$  inputs  $(y_{ID,1}, y_{ID,2}) \leftarrow D_\sigma^m$ , the data file  $F$  and three vectors  $t_1, t_2, t_3$ , the challenger  $\mathcal{C}$  randomly selects  $v \in \{-1, 0, 1\}^m$  and sends it to  $\mathcal{A}$ . Adding  $(F, (y_{ID,1}, y_{ID,2})_{1 \leq ID \leq N}, v, t_1, t_2, t_3)$  to the Hash query list.

*KeyGen query.* Given an identity  $ID \in [1, N]$ , the challenger  $\mathcal{C}$  runs the key extraction algorithm to obtain  $x_{ID,1}, x_{ID,2}$ .

*SignGen query.* The adversary  $\mathcal{A}$  inputs  $ID$  and the data file  $F$ . The challenger  $\mathcal{C}$  sets  $z_{ID,1} = y_{ID,1}, z_{ID,2} = y_{ID,2}$ , and returns  $((z_{ID,1}, z_{ID,2}), w_{ID}^{name}, t_1, t_2, t_3)$ . Adding  $(ID, F)$  to the SignGen query list.

Finally, the adversary  $\mathcal{A}$  forges the signature  $((z_{ID,1}^*, z_{ID,2}^*), (w^*)_{ID,j}^{name}, t_1^*, t_2^*, t_3^*)$  about the file  $F$  and the group member  $ID$  to satisfy the equation. The identity  $ID$  and the file  $F$  are not in the private key challenge list and signature challenge list. From the bifurcation lemma we know, The adversary  $\mathcal{A}$  outputs two signatures  $((z_{ID,1}^*, z_{ID,2}^*), (w^*)_{ID,j}^{name}, t_1^*, t_2^*, t_3^*)$  and  $((\tilde{z}_{ID,1}^*, \tilde{z}_{ID,2}^*), (\tilde{w}^*)_{ID,j}^{name}, t_1^*, t_2^*, t_3^*)$  with probability  $\varepsilon(\frac{\varepsilon}{h_1} - 2^{-n})$  at least, and  $z_{ID,1}^* \neq$

$\tilde{z}_{ID,1}^*, z_{ID,2}^* \neq \tilde{z}_{ID,2}^*, (w^*)_{ID}^{name} = (\tilde{w}^*)_{ID}^{name}$ , where  $h_1 \geq 2$  is the length of the output of the function  $H_4$ .

Since the correctness of signature forged by adversary, the following equation holds:

$$\begin{aligned}
& \sum_{ID=1}^N (Bz_{ID,1}^* + A_{ID} z_{ID,2}^* - u(w^*)_{ID,j}^{name}) = \sum_{ID=1}^N (By_{ID,1} + A_{ID} y_{ID,2}), \\
& \sum_{ID=1}^N (A_{ID} z_{ID,2}^* - u(w^*)_{ID,j}^{name} + g_{ID} (w^*)_{ID,j}^{name}) = \sum_{ID=1}^N A_{ID} y_{ID,2}, \\
& \sum_{ID=1}^N (B\tilde{z}_{ID,1}^* + A_{ID} \tilde{z}_{ID,2}^* - u(\tilde{w}^*)_{ID,j}^{name}) = \sum_{ID=1}^N (By_{ID,1} + A_{ID} y_{ID,2}), \\
& \sum_{ID=1}^N (A_{ID} \tilde{z}_{ID,2}^* - u(\tilde{w}^*)_{ID,j}^{name} + g_{ID} (\tilde{w}^*)_{ID,j}^{name}) = \sum_{ID=1}^N A_{ID} y_{ID,2}.
\end{aligned}$$

The challenger  $\mathcal{C}$  simplifies the above formula, we can get:

$$\begin{aligned}
& \sum_{ID=1}^N (B(z_{ID,1}^* + \tilde{z}_{ID,1}^*) + A_{ID} (z_{ID,2}^* + \tilde{z}_{ID,2}^*)) \\
&= u((w^*)_{ID,j}^{name} - (\tilde{w}^*)_{ID,j}^{name}), \\
& \sum_{ID=1}^N (A_{ID} (z_{ID,2}^* - \tilde{z}_{ID,2}^*)) \\
&= (u((w^*)_{ID,j}^{name} - (\tilde{w}^*)_{ID,j}^{name}) - g_{ID}((w^*)_{ID,j}^{name} - (\tilde{w}^*)_{ID,j}^{name})).
\end{aligned}$$

Thereby the equation (14) holds. Because  $(w^*)_{ID,j}^{name} \neq (\tilde{w}^*)_{ID,j}^{name}$  gets. Thus the equation (15) holds.  $\sum_{ID=1}^N (B(z_{ID,1}^* + \tilde{z}_{ID,1}^*)) = -g_{ID}((w^*)_{ID,j}^{name} - (\tilde{w}^*)_{ID,j}^{name})$  (14)  $\sum_{ID=1}^N (B(x_{ID,1} (w^*)_{ID,j}^{name} = 0 \bmod q)$  (15)

Furthermore, since  $\|\sum_{ID=1}^N (x_{ID,1} (w^*)_{ID,j}^{name})\|_\infty \leq \xi$  holds, the challenger  $\mathcal{C}$  uses the fake result of the adversary  $\mathcal{A}$  to construct a solution to the RSIS problem.

*Theorem 2* (Full traceability). Our scheme is full traceability under the random oracle model, if the RSIS problem is hard.

*Proof.* Assuming that the adversary  $\mathcal{A}$  successfully forges untraceable signatures with a non-negligible probability, the challenger  $\mathcal{C}$  uses the forged results of the adversary  $\mathcal{A}$  to construct a set of non-zero solutions to the RSIS problem. The challenger  $\mathcal{C}$  set the initialization list  $L_1, L_2, L_3$  to empty and enter the security parameter  $n$  and the maximum number of users  $\{ID\}_1^N$  in the group. The challenger  $\mathcal{C}$  randomly chooses an  $ID' \in [1, N]$ , runs the algorithm honestly and gets  $gtk_j, gmk_j, gsk_j$ . The challenger sends  $gmk_j$  and  $gtk_j$  to adversary  $\mathcal{A}$ . The adversary  $\mathcal{A}$  interacts with the challenger  $\mathcal{C}$  as follows:

*Private key query.* Given an identity  $ID^* \in [1, N]$ . If  $ID^* = ID'$ , the game is terminated; otherwise, if  $ID^* \neq ID'$ , the challenger  $\mathcal{C}$  returns the corresponding private key  $gsk_{ID^*}$  to the adversary  $\mathcal{A}$ . The challenger  $\mathcal{C}$  saves  $ID^*$  to list  $L_1$ .

*SignGen query.* Given an identity  $ID \in [1, N]$  and a data file  $F$ , the challenger  $\mathcal{C}$  sets  $z_{ID,1} = y_{ID,1}, z_{ID,2} = y_{ID,2}$ , returns signature  $((z_{ID,1}, z_{ID,2}), w_{ID,j}^{name}, t_1, t_2, t_3)$  and appends  $(ID, F)$  into list  $L_2$ .

*Hash query.* Given two vectors  $(y_{ID,1}, y_{ID,2}) \leftarrow D_\sigma^m$ , a data file  $F$  and  $t_1, t_2, t_3$ . The challenger  $\mathcal{C}$  check list  $L_3$ , if the adversary  $\mathcal{A}$  has asked, return the result in the list; otherwise, the challenger  $\mathcal{C}$  randomly selects  $v \in \{-1, 0, 1\}^m$  and sends it to the adversary  $\mathcal{A}$ . The challenger  $\mathcal{C}$  appends

$(F, (y_{ID,1}, y_{ID,2})_{1 \leq ID \leq N}, v, t_1, t_2, t_3)$  into list  $L_3$ .

Finally, the adversary  $\mathcal{A}$  outputs the signature. Assume that the signature is a valid signature, satisfying

$$\begin{aligned} w_{ID,j}^{name} &= H_5(\sum_{ID=1}^N (By_{ID,1} + \\ A_{ID} y_{ID,2}), \sum_{r=1}^l f_{ID,r,j}^{name}, t_1, t_2, t_3), \\ (w^*)_{ID,j}^{name} &= H_5(\sum_{ID=1}^N (By_{ID,1} + \\ A_{ID} y_{ID,2}), \sum_{r=1}^l f_{ID,r,j}^{name}, t_1^*, t_2^*, t_3^*). \end{aligned}$$

It can be obtained  $t_1 = t_1^*$ ,  $t_2 = t_2^*$  and  $t_3 = t_3^*$ . At this time, the equation  $\sum_{ID=1}^N (B(y_{ID,1} - y_{ID,1}^*) + A_{ID}(y_{ID,2} - y_{ID,2}^*)) = 0 \bmod q$  is established. Because of  $\|\sum_{ID=1}^N (y_{ID,1} - y_{ID,1}^*)\|_\infty \leq \xi$ ,  $\sum_{ID=1}^N (y_{ID,1} - y_{ID,1}^*)$  is the solution to the RSIS problem.

**Theorem 3** (Conditional Anonymity). Our scheme is conditional anonymity under the random oracle model, if the RLWE problem is hard.

*Proof.* Construct the following games to prove the conditional anonymity of the scheme.

**Game1:** The challenger  $\mathcal{C}$  enters the security parameters  $n$  and the maximum number of group members  $N$  and honestly runs the *Setup* algorithm to generate the master key, the traceability key and the public parameters for group  $j$  are  $gsk_j, gtk_j, pp$ . The challenger  $\mathcal{C}$  sends  $pp, gtk_j$  to the adversary  $\mathcal{A}$ . Then, the challenger  $\mathcal{C}$  runs the *NewBasisDel* algorithm to generate the user's private key  $gsk_{ID}$  and send it to adversary  $\mathcal{A}$ . The adversary  $\mathcal{A}$  uses  $\{ID', F', \psi'\}$  to perform *SignGen* query and *Open* query:

*SignGen query.* The adversary  $\mathcal{A}$  enters  $ID_1, ID_2 \in [1, N]$ ,  $ID_1 \neq ID_2$  and the file  $F$ . The challenger  $\mathcal{C}$  randomly selects the identity index  $b \in \{0, 1\}$  and runs the *KeyGen* algorithm to generate the user's signature private key  $gsk_b$ . Finally, the challenger  $\mathcal{C}$  returns the signature  $\psi = ((z_{b,1}, z_{b,2}), w_{b,j}^{name}, t_1, t_2, t_3)$ .

*Open query.* The adversary  $\mathcal{A}$  enters the file  $F$  and the corresponding signature  $\psi$ , and if the signature is valid, returns the user identity  $ID$ , otherwise terminates.

The adversary  $\mathcal{A}$  provides the file  $F$  and two valid identities  $1 \leq ID_1, ID_2 \leq N$ , and the challenger  $\mathcal{C}$  gets the signature  $\psi = ((z_{b,1}, z_{b,2}), w_{b,j}^{name}, t_1, t_2, t_3)$ . The adversary  $\mathcal{A}$  outputs a guess  $b' \in \{0, 1\}$  about the identity  $ID_b$  of the signer. Then, the probability that  $b' = b$  is  $Pr(G_1) = 1/2 + \text{negl}(n)$ .

**Game2:** From the PLR double encryption algorithm, the adversary  $\mathcal{A}$  can only obtain valid information about the signer from  $t_2$  and  $t_3$ . Thus, it is easy to verify that the probability of the Game1 succeeding is equal to the probability of the adversary  $\mathcal{A}$  breaking the verification scheme.

**Game3:** The system building phase is the same as in Game1, where the adversary  $\mathcal{A}$  makes a *SignGen* query. The adversary  $\mathcal{A}$  inputs the files  $F_1$  and  $F_2$ , then, the challenger  $\mathcal{C}$  runs the *SigGen* algorithm with  $F_b$  and selects two randomly files, returning signatures  $\psi_1 = ((z_{ID,1}, z_{ID,2}), w_{ID,j}^{F_1}, t_1, t_2, t_3)$  and  $\psi_2 = ((z_{ID,1}, z_{ID,2}), w_{ID,j}^{F_2}, t_1, t_2, t_3)$ . Finally, the adversary  $\mathcal{A}$  outputs a guess  $b' \in \{0, 1\}$  about the message  $F_b$ . Therefore, the probability that the adversary  $\mathcal{A}$  guesses  $b' = b$  in the Game 3 can be calculated as  $Pr(G_2) = 1/2 + \text{negl}(n)$ .

From Game1, Game2 and Game3, we can be obtained that

the advantage of conditional anonymity of the adversary  $\mathcal{A}$  breach of this scheme is negligible.

## 5 Performance evaluation

### A. Functionality

In this section, we present a functionality comparison between our scheme and existing related schemes [13, 16, 19] in terms of anonymity, traceability, multiples group, user revocation, batch auditing and resist quantum attack in Table 1.

Scheme [13] is a dynamic data integrity auditing scheme in untrustworthy cloud servers with a static group. Scheme [16] is a group data integrity auditing scheme without group manager, and scheme [19] is an integrity auditing scheme for group management of shared cloud data. The above three schemes [13, 16, 19] achieve anonymity and traceability, and scheme [16] realizes user revocation. What a pity, all of the above schemes [13, 16, 19] only discusses data sharing among members of a single group, and do not consider the issue of outsourced data sharing among multiple groups. Furthermore, the above schemes do not fulfill batch auditing. Meanwhile, our scheme not only solves the data sharing problem well based on multiple groups, but also achieves resistance to quantum attacks. Thus, our proposed scheme has more comprehensive features and has a wider range of applications than schemes [13, 16, 19].

**Table 1** Functionality Comparison

| Scheme                 | [13] | [16] | [19] | Ours |
|------------------------|------|------|------|------|
| Anonymity              | ✓    | ✓    | ✓    | ✓    |
| Traceability           | ✓    | ✓    | ✓    | ✓    |
| Multiples Group        | ×    | ×    | ×    | ✓    |
| User Revocation        | ×    | ✓    | ×    | ✓    |
| Batch Auditing         | ×    | ×    | ×    | ✓    |
| Resist quantum attacks | ×    | ×    | ×    | ✓    |

### B. Communication Cost

In this section, our proposed scheme is compared with schemes [19] and [26] in terms of communication cost as shown in Table 3. For the convenience of comparison, the main symbols and their meanings are shown in Table 2.

The communication cost of our proposed scheme mainly comes from the auditing challenge and auditing proof in the auditing process. In the challenge generation phase, the TPA sends the challenge information  $chal = (i, \{h_i\}_{i \in I})$  to the CSP.

**Table 2** Symbols and their meaning

| Symbol     | Meaning                                                           |
|------------|-------------------------------------------------------------------|
| $c_r$      | The number of RU-blocks                                           |
| $T_{Hash}$ | The time cost of performing once hash function                    |
| $c$        | The number of the challenged block                                |
| $T_{Add}$  | The time cost of performing once addition on $G_1$                |
| $T_{exp}$  | The time cost of exponentiation on $G_1$                          |
| $T_{pair}$ | The time cost of bilinear pairings on $G_1$                       |
| $T_{mult}$ | The time cost of performing once multiplication on vectors, $G_1$ |

The communication cost TPA in our scheme and scheme [26] both is  $c(|l| + |q|)$ . So the two schemes have similar communication cost on the TPA side. In scheme [19], the challenge generation phase, the TPA not only generates challenge messages for the non-revoked users, but also generates messages for the revoked users and forwards them to the CSP. Therefore, the communication cost of the TPA in scheme [19] is  $(c + c_r)(|l| + |q|)$ , which is higher than our proposed scheme. In the proof generation phase, the communication cost of CSP in scheme [19] is independent of the number of users and the communication cost is the lowest. The communication cost of CSP in scheme [26] is  $c|ID| + |G_2|$ , which is related to the length of user identity and the number of group members. The communication cost of CSP in our proposed scheme is  $(Nm + n)|q|$  related to the number of group members. Our proposed scheme contains multiple groups, thus, the CSP needs to save the information of all users in the group, resulting in a high communication cost on the CSP side. However, the CSP has a strong communication capability, which does not affect the efficiency of the scheme as long as the number of group members is within a certain range. In this sense, our scheme not only completes the batch verification of data blocks but also multiple groups, so it is more efficient.

**Table 3** Comparison of Communication Cost

| Schemes    | TPA                    | CSP             |
|------------|------------------------|-----------------|
| [19]       | $(c + c_r)( l  +  q )$ | $ q $           |
| [26]       | $c( l  +  p )$         | $c ID  +  G_2 $ |
| Our scheme | $c( l  +  q )$         | $(Nm + n) q $   |

### C. Computational Cost

In this section, we further evaluate the computational cost of our scheme and schemes [19, 26] as shown in Table 4.

In the proof generation stage, schemes [19, 26] have computational costs respectively  $(c_r + c + 1)T_{pair} + 2cT_{Add}$  and  $c(n + 1)T_{pair} + c|T_{Add}_{exp}$ . Therefore, compared with schemes [19, 26], our proposed scheme has a minimum computational cost  $cT_{mult}$ . In the verification phase, the computational cost of scheme [19] is  $nT_{Hash} + T_{Add} + T_{exp}$ , and the computational cost of scheme [26] is  $cT_{pair}_{Hash_{exp}}$ . The computational cost of our scheme is  $(m + N)T_{Hash} + cT_{mult}$ . The computational complexity of the hash function is of constant order, so the  $m + N$  hash operation does not increase the computational cost. In addition, the simple modulo multiplication operation used in the verification phase of our scheme reduces the computational overhead of the auditor. Therefore, compared with existing schemes [19, 26], our scheme achieves resistance to quantum attacks with higher efficiency and practicality.

**Table 4** Comparison of Computational Cost

| Schemes | Proof Generation                     | Verifying                       |
|---------|--------------------------------------|---------------------------------|
| [19]    | $(c_r + c + 1)T_{pair} + 2cT_{Add}$  | $nT_{Hash} + T_{Add} + T_{exp}$ |
| [26]    | $c(n + 1)T_{pair} + c T_{Add}_{exp}$ | $cT_{pair}_{Hash_{exp}}$        |
| Ours    | $cT_{mult}$                          | $(m + N)T_{Hash} + cT_{mult}$   |

## 6 Conclusion

Compared with other algorithms, lattice-based group signature is resistant to quantum algorithm attacks, and identity-based group signature on lattices can avoid complicated certificate management problems. Our proposed scheme uses rejection sampling algorithms to improve the efficiency of signatures and ensure forward security and confidentiality of data through user revocation. In the data auditing phase, batch validation of data is completed in three different cases and the auditing process is described in detail. This procedure can be adopted when multiple organizations, such as telemedicine systems, share outsourced data among multiple medical groups to improve the research level of medical groups and help healthcare workers improve the accuracy of judgments.

**Acknowledgments** The work was supported by Foundation Items: The National Natural Science Foundation of China (No. 62262060, 61662071, 62241207); Industrial support plan project of Gansu Provincial Department of Education (2022CYZC-17); Gansu Science and Technology Program (22JR5RA158).

**Author's contribution** Xueyan Liu and Xiaoyan Zhang wrote the manuscript and revised the paper, Jia Wang and Xiliang Li supervised and analyzed the scheme.

**Data availability** We also declare that data availability and ethics approval is not applicable in this paper.

**Statements and Declarations** The authors declare that they have no conflict of interest.

## References

- [1] Su Q, Zhang R, Xue R, and Li P. (2020) "Revocable attribute-based signature for blockchain-based healthcare system[J]." IEEE Access, vol. 8, pp. 127884-127896, 2020. DOI: <https://doi.org/10.1109/ACCESS.2020.3007691>.
- [2] Shen W, Qin J, Yu J, Hao R, and Hu J. (2018) "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage[J]." IEEE Transactions on Information Forensics and Security, vol. 14, no. 2, pp. 331-346, 2018. DOI: <https://doi.org/10.1109/TIFS.2018.2850312>.
- [3] Garg N, and Bawa S. (2016) "Comparative analysis of cloud data integrity auditing protocols[J]." Journal of network and computer applications, vol. 66, pp. 17-32, 2016. DOI: <https://doi.org/10.1016/j.jnca.2016.03.010>.
- [4] Ravishanker B, Kulkarni P, and Vishnudas M. V. (2020) "Blockchain-based database to ensure data integrity in cloud computing environments[J]." 2020 International Conference on Mainstreaming Block Chain Implementation (ICOMBI). IEEE, 2020. DOI: <https://doi.org/10.23919/ICOMBI48604.2020.9203500>.
- [5] Shao B, Bian G, Wang Y, Su S, and Guo C. (2018) "Dynamic data integrity auditing method supporting privacy protection in vehicular cloud environment[J]." IEEE Access, vol. 6, pp. 43785-43797, 2018. DOI: <https://doi.org/10.1109/ACCESS.2018.2863270>.
- [6] Jegadeesan R and Sahithi C. (2021) "A Scalable Mechanism of Cloud Storage for Data Integrity Auditing without Private Key Storage[J]." International Journal of Research, vol. 10, pp. 1-8, 2021. DOI: <https://doi.org/10.13140/RG.2.2.34954.59845>.
- [7] Han H, Fei S, Yan Z, and Zhou X. (2022) "A survey on blockchain-based integrity auditing for cloud data[J]." Digital Communications and Networks, 2022. DOI: <https://doi.org/10.1016/j.dcan.2022.04.036>.

- [8] Shen J, Shen J, Chen X, Huang X, and Susilo W. (2017) "An efficient public auditing protocol with novel dynamic structure for cloud data[J]." *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402-2415, 2017. DOI: <https://doi.org/10.1109/TIFS.2017.2705620>.
- [9] Zhao Q, Chen S, Liu Z, Baker T, and Zhang Y. (2020) "Blockchain-based privacy-preserving remote data integrity checking scheme for IoT information systems[J]." *Information Processing & Management*, vol. 5, no. 6, pp. 102355, 2020. DOI: <https://doi.org/10.1016/j.ipm.2020.102355>.
- [10] Xie R, He C, Xu C, and Gao C. (2019) "Lattice-based dynamic group signature for anonymous authentication in IoT[J]." *Annals of Telecommunications*, vol. 74, no. 7, pp. 531-542, 2019. DOI: <https://doi.org/10.1007/s12243-019-00705-x>.
- [11] Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, and Song D. (2007) "Provable data possession at untrusted stores[C]." *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 598-609, 2007. DOI: <https://doi.org/10.1145/1315245.1315318>.
- [12] Liu X, Luo Y, Yang X, Wang and Zhang X. (2021) "Lattice-Based Proxy-Oriented Public Auditing Scheme for Electronic Health Record in Cloud-Assisted WBANs[J]" *IEEE Systems Journal*, vol. 16, no. 2, pp. 2968-2978. DOI: <https://doi.org/10.1109/JSYST.2021.3138861>.
- [13] Shen W, Qin J, Yu J, Hao R, Hu J, and Ma J. (2019) "Data integrity auditing without private key storage for secure cloud storage[J]" *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1408-1421, 2019. DOI: <https://doi.org/10.1109/TCC.2019.2921553>.
- [14] Yang C, Song B, Ding Y, Ou J, and Fan C.(2022) "Efficient data integrity auditing supporting provable data update for secure cloud storage[J]." *Wireless Communications and Mobile Computing*, 2022. DOI: <https://doi.org/10.1155/2022/5721917>.
- [15] Guo W, Qin S, Gao F, Zhang H, Li W, et al.(2020) "Dynamic Proof of Data Possession and Replication With Tree Sharing and Batch Verification in the Cloud[J]" *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 1813-1824. DOI: <https://doi.org/10.1109/TSC.2020.3022812>.
- [16] Shang T, Zhang F, Chen X, Liu J and Lu X. (2019) "Identity-Based Dynamic Data Auditing for Big Data Storage[J]" *IEEE Transactions on Big Data*, vol. 7, no. 6, pp.913-921, 2021. DOI: <https://doi.org/10.1109/TBDATA.2019.2941882>.
- [17] Garg N, Bawa S, and Kumar N. (2020) "An efficient data integrity auditing protocol for cloud computing[J]" *Future Generation Computer Systems*, vol. 109, pp. 306-316, 2020. DOI: <https://doi.org/10.1016/j.future.2020.03.032>.
- [18] Wang B, Li B and Li H. (2014) "Oruta: privacy-preserving public auditing for shared data in the cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 43-56, 2014. DOI: <https://doi.org/10.1109/TCC.2014.2299807>.
- [19] Tian H, Nan F, Jiang H, Chang C, Ning J and Huang Y. (2019) "Public auditing for shared cloud data with efficient and secure group management[J]" *Information Sciences*, vol. 472, pp. 107-125, 2019. DOI: <https://doi.org/10.1016/j.ins.2018.09.009>.
- [20] Shen J, Zhou T, Chen X, Li J and Susilo W. (2018) "Anonymous and Traceable Group Data Sharing in Cloud Computing[J]" *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912-925, 2018. DOI: <https://doi.org/10.1109/TIFS.2017.2774439>.
- [21] Lyubashevsky V, Peikert C, and Regev O. (2010) "On ideal lattices and learning with errors over rings[C]." *Annual international conference on the theory and applications of cryptographic techniques*. Springer, Berlin, Heidelberg, vol. 6110, pp. 1-23, 2010. DOI: [https://doi.org/10.1007/978-3-642-13190-5\\_1](https://doi.org/10.1007/978-3-642-13190-5_1).
- [22] Agrawal S, Boneh D, and Boyen X. (2010) "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE[C]." *Annual cryptography conference*. Springer, Berlin, Heidelberg, vol. 6223, pp. 2010. DOI: [https://doi.org/10.1007/978-3-642-14623-7\\_6](https://doi.org/10.1007/978-3-642-14623-7_6).
- [23] Lyubashevsky V. (2012) "Lattice signatures without trapdoors[C]." *Annual International Conference on the Theory and Applications of Cryptosystem Techniques*. Springer, Berlin, Heidelberg, vol. 737, pp. 738-755, 2012. DOI: [https://doi.org/10.1007/978-3-642-29011-4\\_43](https://doi.org/10.1007/978-3-642-29011-4_43).
- [24] Micciancio D. (2007) "Generalized compact knapsacks, cyclic lattices, and efficient one-way functions[C]." *Computational complexity*, vol. 16, no. 4, pp. 365-411, 2007. DOI: <https://doi.org/10.1007/s00037-007-0234-9>.
- [25] Peikert C and Rosen A. (2006) "Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices[C]" *Theory of Cryptography Conference*. Springer, Berlin, Heidelberg, vol. 3876, pp. 145-166, 2006. DOI: [https://doi.org/10.1007/11681878\\_8](https://doi.org/10.1007/11681878_8).
- [26] Thokchom S, and Saikia D. K. (2018) "Efficient scheme for dynamic cloud data shared within a static group with privacy preserving auditing and traceability[C]." *Proceedings of the 2018 International Conference on Cloud Computing and Internet of Things*. pp. 25-32, 2018. DOI: <https://doi.org/10.1145/3291064.3291072>.