Revocable Attribute-Based Data Integrity Auditing Scheme on Lattices

Xiaoyan Zhang[1*], Xueyan Liu[2], Qiong Liu[2], Jing Wang[2]

[1] College of Mathematics and Statistics, Northwest Normal University, Lanzhou, 730070, China
[2] College of Computer Science and Engineering, Northwest Normal University, Lanzhou, 730070, China

Email: Xiaoyan ZHANG, zxiaoyan97@163.com
Xueyan LIU, liuxy@nwnu.edu.cn
Qiong LIU, 593095942@qq.com
Jing WANG, 1219980108@qq.com
*Corresponding author: Xiaoyan ZHANG

Abstract. To save local storage space and protect data privacy, enterprises store data in the private cloud. The private cloud is only open to internal users of the enterprise, but the data owners lose control over outsourced data, so data integrity auditing is still an urgent problem to be solved in secure cloud storage. Although the existing data integrity auditing schemes on lattices can effectively resist the quantum attack, they lack fine-grained management of user permissions and have key escrow problems. This paper combines attribute-based signature on lattices to construct a revocable attribute-based data integrity auditing scheme on lattices. Firstly, the system master key is generated by using the trapdoor generation algorithm, and then combined with the user's attribute set, the user's initial key is generated by using the lattice extbasis algorithm. Secondly, users add their own identity information and generate real keys without key escrow by using lattice randbasis algorithm. In the data submission stage, the Gauss sampling algorithm and the lattice extbasis algorithm are combined to generate the signature. Moreover, the authorization center periodically updates the revocation list through user identities and attribute sets to implement dynamic management of users. Based on the hardness assumption of SIS problem, it is proved that the scheme has strong unforgeability and storage correctness. Compared with the existing data integrity auditing scheme, the security and practicability are higher.

Keywords: private cloud; integrity verification; lattice-based cryptography; post-quantum security; user revocation

1 Introduction

In the digital age of information, data is growing exponentially and both individuals and enterprises are facing huge data storage problems. Cloud storage relieves the pressure of local data storage and maintenance, and brings great convenience to data storage

H. Wu et al. (Eds.): CSIEDE 2022, ACSR 103, pp. 383–396, 2022.
https://doi.org/10.2991/978-94-6463-108-1_44

and access. While cloud storage provides convenient services to users, it also raises some security issues. Such as the interruption of public cloud servers, various attacks existing on the network, and personal or corporate privacy data leakage. Therefore, to enhance the privacy protection of enterprise data, most enterprises establish their private clouds to store enterprise data.

Private clouds can guarantee enterprise confidentiality and data security storage. It not only solves the problem of large amounts of data storage, but also strengthens the protection of data confidentiality. However, its storage scale and service capacity cannot match that the public cloud, so only internal users can upload or read the data in the private cloud. Generally, in the enterprise, although employees can upload or read the data in the private cloud, the permission division is relatively clear. For example, the data of the finance department can only be charged by the data manager of the finance department, and the content of the department managers' meetings should not be accessed by the general staff, etc. In addition, when the department of the employee is transferred or the employee resigns, the employee's permission will be changed or revoked, so that the departments can avoid cross accessing and disseminating enterprise data. Therefore, most of the currently existing data integrity auditing schemes in public clouds have problems such as the insufficient granularity of user permission management and lack of dynamic user management, which are not suitable for the data integrity auditing requirements in private clouds. Therefore, it is necessary to explore data integrity auditing schemes in private clouds that can manage fine-grained user permissions and dynamic user management.

2 Related work

With the development of cryptographic techniques, many cloud storage data integrity auditing schemes have been proposed [1], [2], [3], [4]. Data integrity auditing schemes are mainly divided into two categories, provable data possession (PDP) [5] and proof of retrievability (POR) [6]. Cloud storage data integrity auditing schemes that support public auditing allow to delegate the integrity auditing of outsourced data to professional third-party auditors to reduce the burden of user auditing. To prevent curious third-party auditors from accessing user information to disclose user privacy, Zhang et al. [7] proposed an auditing scheme for electronic medical data in cloud-assisted WBANs with designated verifiers. The key generation center (KGC) generates private keys based on the user's identity and designates verifiers to check the integrity of medical data in the cloud server. However, identity-based cryptosystems have key escrow issues. Yu et al. [8] proposed the concept of attribute-based cloud data integrity auditing. The KGC generates private keys to users based on attribute sets to achieve fine-grained management of user permissions and simplify the key management problem of identity-based cryptosystems. In the scheme8, the KGC is required to be fully trusted and the KGC may forge signatures to leak data. Wang et al. [9] proposed an attribute-based data integrity auditing scheme supporting user revocation. The group manager sends secret information to the third-party auditors to track the signer. User revocation is achieved by revoking the user's access to the data when the user's attributes are changed. However, in this scheme, unauthorized third-party auditors can also send auditing requests to the cloud server, which increases the communication burden of the cloud server.

With the development of quantum computers, the above schemes cannot resist quantum attacks. Subsequently, Li et al. [10] proposed a certificateless public auditing scheme for post-quantum security in cloud storage, where the user private key is jointly generated by KGC and the user to solve the key escrow problem. Liu et al. [11] proposed a lattice-based proxy-oriented public auditing scheme for electronic health records with cloud assistance, where the user authorizes the proxy to complete the signature of the data block and the user blinds data before sending it to the proxy. To protect the privacy of the original data and reduce the computational overhead on the user side. However, schemes [10] and [11] do not have fine-grained management of user permissions.

In summary, most existing data integrity auditing schemes are based on traditional cryptographic difficulties such as large integer factorization and discrete logarithm, and cannot resist quantum attacks. Secondly, current research tends to focus on the data integrity auditing requirements of data managers, ignoring the need that data belonging to companies and data managers need to be able to be flexibly replaced for various reasons.

Therefore, this paper proposes a revocable attribute-based data integrity auditing scheme on lattice. The main contributions are as follows.

1) Resisting quantum attacks and fine-grained management of user permissions. A data integrity auditing scheme on lattice is constructed by using lattice expansion algorithm, lattice randomization algorithm and other technologies. Based on the hardness assumption of small integer solution (SIS) problem, it is resistant to quantum attacks. The user private key is generated by combining the user attribute set, and the signature generation stage uses the linear secret sharing scheme (LSSS) to achieve fine-grained management of user permission.

2) User dynamic management. When a user resigns or his position is adjusted, the authorization center revokes the user who leaves and reauthorizes the user who changes his position. When the user's attribute set and identity information do not match the information in the private cloud, the user can no longer read or modify the data previously uploaded to the private cloud, to ensure the forward security of data and realize dynamic management of users.

3) Low computational overhead and efficient verification. Compared with existing data integrity auditing schemes, the proposed scheme has stronger practicality in resisting quantum attacks without the need to perform complex and time-consuming bilinear pair operations as well as modulo-exponential operations. To address the efficiency bottleneck that occurs on the auditor side in some specific applications, in the verification phase, the auditor constructs the verification matrix based on the LSSS and only performs simple modulo addition and modulo multiplication operations to reduce the computational overhead of the auditor.

3 Preliminaries

3.1 Lattices

Definition 1: Let $B=\{b_1,b_2,\cdots b_n\}\in R^{m\times m}$ be a set of linearly independent column vectors, it generates an m-dimensional full-rank lattice Λ, which is defined as $\Lambda=\{Bc=\sum\limits_{1\le i\le m} c_i b_i, c_i\in Z\}$.

3.2 Hardness assumption

Definition 2: The Small Integer Solution problem (SIS) is described below. Given a prime q, a real number $\xi>0$ and a matrix $A\in\mathbb{Z}_q^{n\times m}$ and a vector $y\in\mathbb{Z}_q^n$, to solve a nonzero integer vector $e\in\mathbb{Z}^m$ such that $Ae=y$ and $\| e \|\le\xi$ [12].

3.3 System model

The system model contains four entities: the authorization center (AC), the user, the third-party auditor (TPA), and the private cloud (CSP). The system model is shown in Figure 1. The functions of each entity in the system framework are as follows.

AC. Responsible for giving authorization to users as well as auditors and generating public and master keys as well as completing user revocation, but it is not full trusted.

User. Users, in the enterprise who generate work data, are responsible for uploading or updating their work data to the private cloud.

TPA. A third-party auditor within the company that has powerful computing power and is responsible for regularly checking the integrity of the cloud data.

CSP. It is the enterprise private cloud. Hold data files uploaded by internal users in the enterprise with legal permission and have great storage space.

4 The proposed scheme

In this paper, we introduce a non-fully trusted authorization center, the user runs the lattice-based randomization algorithm during the key extraction process, and the authorization center cannot fully obtain the signature private key, effectively preventing the authorized authority from forging the signature. Combined with attribute-based signatures and using LSSS access structure, the user authorization method is more flexible and fine-grained. In addition, private clouds and auditors judge user permissions based on user identity and attributes, ensuring user legitimacy and data privacy. The global attribute sets are $Att=\{Att_1,Att_2,\cdots,Att_K\}$, $Att\in\mathbb{Z}_q^n$, the total number of attribute sets is K, the user attribute set is $U=\{U_{att_1},\cdots,U_{att_k}\}$, the total number of user attribute sets is k, and the user identity is $ID\in\{0,1\}^*$.

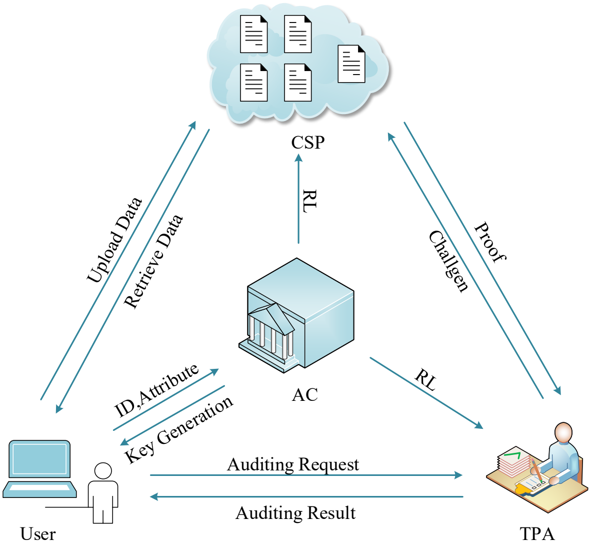

Fig. 1. System model [Owner-draw]

Setup: The system sets security parameters n , $q = ploy(n)$, $m = O(n\log q)$, $m_1, m_2 = Cn\log q$ and picks Gaussian parameters $s = mw(\log n)$, where C is a constant greater than 1. Then, the system establishment algorithm consists of the following:

(a) The system runs the $TrapGen(n, m, q)$ algorithm to generate a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m_1}$ and a short basis $\boldsymbol{T}_A \in \mathbb{Z}_q^{m_1\times m_1}$ of $\Lambda_q^{\perp}(\boldsymbol{A})$.

(b) The system randomly selects r matrices $\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots, \boldsymbol{C}_r \in \mathbb{Z}_q^{n\times m_2}$.

(c) The system randomly selects k uniform random matrices $\boldsymbol{A}_i \in \mathbb{Z}_q^{n\times m_2}$, $i = 1, 2, \cdots, k$.

(d) The system defines four hash functions: $H_1 : \{0,1\}^* \to \mathbb{Z}_q^{m_1\times m_1}$, $H_2 : \mathbb{Z}_q^{n\times(m_1+km_2)} \to \mathbb{Z}_q^{n\times n}$, $H_3 : \{0,1\}^* \times \{0,1\}' \times [1\times n] \to \mathbb{Z}_q^n$, $H_4 : \mathbb{Z}_q^{n\times m_2} \to \mathbb{Z}_q^n$.

(e) Finally, the system outputs a revocation list RL , public parameters $pp = \{\boldsymbol{A}, \{\boldsymbol{A}_i\}_{i\in Att}, \{\boldsymbol{C}_i\}_{i\in[1,l]}, s, H_1, H_2, H_3, H_4\}$ and master secret key $MSK = \boldsymbol{T}_A$.

Extract: Inputs the master secret key $\boldsymbol{T}_A$, the system public parameters pp , the set of user attributes and user's identity $ID \in \{0,1\}^*$, the AC and the user generates the user's private key.

(a) Let the set of user attributes be $U = \{U_{att_1}, \cdots, U_{att_k}\}$. If $U_{att_i} = Att_i$, then let $U_{att_i} = \boldsymbol{A}_i$, otherwise $U_{att_i} = 0$. Obtain the attribute matrix as $\bar{\boldsymbol{A}}_u = (\boldsymbol{A} \mid \boldsymbol{A}_1 \mid \boldsymbol{A}_2 \mid \cdots \mid \boldsymbol{A}_k) \in \mathbb{Z}_q^{n\times(m_1+km_2)}$. The AC runs $ExBasis(\boldsymbol{A}, \bar{\boldsymbol{A}}_u, \boldsymbol{T}_A, s)$ algorithm to obtain a random short basis $\bar{\boldsymbol{T}}_u$ of lattice $\Lambda_q^{\perp}(\bar{\boldsymbol{A}}_u)$ and sends it to the user through a secure channel. The AC computes $\boldsymbol{R}_u = H_2(\bar{\boldsymbol{A}}_u) \in \mathbb{Z}_q^{n\times n}$, $\boldsymbol{P}_u = \boldsymbol{R}_u \| \boldsymbol{R}$ and sends $\boldsymbol{P}_u$ to the TPA and the CSP.

(b) The user calculates $\boldsymbol{R} = H_1(ID) \in \mathbb{Z}_q^{m_1\times m_1}$. By running the $TrapGen$ algorithm, obtains a short basis $\boldsymbol{T_R} \in \mathbb{Z}_q^{m_1\times m_1}$ of $\Lambda_q^{\perp}(\boldsymbol{R})$ as the secret value and lets $\boldsymbol{T} = \boldsymbol{T}_R + \bar{\boldsymbol{T}}_u$, $\boldsymbol{Q} = \boldsymbol{A}\|\boldsymbol{R} \in \mathbb{Z}_q^{n\times 2m_1}$, $\boldsymbol{A}_u = (\boldsymbol{Q} \mid \boldsymbol{A}_1 \mid \boldsymbol{A}_2 \mid \cdots \mid \boldsymbol{A}_k) \in \mathbb{Z}_q^{n\times(2m_1+km_2)}$. Then, the user runs $RandBasis(ExBasis(\boldsymbol{Q}, \boldsymbol{A}_u, \boldsymbol{T}, s), s)$ algorithm to obtain a random short basis $\boldsymbol{T}_{ID}$ of lattice $\Lambda_q^{\perp}(\boldsymbol{A}_u)$ as the corresponding privacy key $\boldsymbol{T}_{ID}$ of ID .

(c) The user public key is $PK_{ID} = \{\boldsymbol{Q}, \{\boldsymbol{A}_i\}_{i=[1,\cdots,k]}\}$.

SignGen: The user divides the file $\mathscr{F}$ into l data blocks and the data file $\mathscr{F}$ is the $m \times l$ matrix. Each file $\mathscr{F}$ has its corresponding name $name \in \{0,1\}^{'}$. $\tilde{\mathrm{F}} = \{f_1, f_2, \cdots, f_l\}$, subfile $f_j \in \mathbb{Z}_q^m$, $1 \le j \le l \le r$, The user calculates $v_j = H_3(ID \,\|\, name \,\|\, j) \in \mathbb{Z}_q^n$.

(a) The user chooses a subset of user attributes $S_u \subseteq U$ satisfying the signature access policy (L, ρ) . From LSSS, there exists a set of constants satisfying $g_i L_i = [1, 0, \cdots, 0]$, where $S_u = \{\mathrm{i} \in [1], \rho(\mathrm{i}) \in U\}$.

(b) The user calculates $ff_j = H_4(\boldsymbol{C}_j^{(f_j)})$, $\boldsymbol{\beta} = ff_j + \boldsymbol{v}_j$ and lets $A_{ff_j} = [\boldsymbol{A}_u \,\|\, \boldsymbol{C}_j^{(f_j)}]$.

(c) The user runs $SampleD(ExBasis(SK_{ID}, \boldsymbol{A}_{ff_j}, \boldsymbol{A}_u, s), \boldsymbol{\beta}, s)$ algorithm to generate the signature σ_j of data block f_j and obtains the set of signatures of file $\mathscr{F}$ as $\psi = \{\sigma_j\}_{1\le j\le l}$. Then, the user uploads attribute sets, identity, and $\{\tilde{F}, \tau, \psi\}$ to the CSP.

After receiving the signature, the CSP verifies the validity of the signature and calculates $H_2(\bar{\boldsymbol{A}}_u)$, $H_1(ID)$ based on the attribute set and identity uploaded by the user, and determines whether equation (1) holds. If it is valid, continue to check whether equation (2) is valid, and if the verification also passes, CSP saves the user's attribute set, identity and $\{\tilde{F}, \tau, \psi\}$. Otherwise, the CSP refuses it.

$$H_2(\bar{\mathrm{A}}_u)||H_1(ID) = \mathrm{P}_u \tag{1}$$

$$\mathrm{A}_u \cdot \sigma_j = \beta \, mod \, q \tag{2}$$

ChalGen: The TPA receives an auditing request from a cloud user with attribute set S_u . Firstly, the TPA checks whether the cloud user's attribute set S_u satisfies the access policy and calculates $H_2(\bar{\boldsymbol{A}}_u)$, $H_1(ID)$ based on the user attributes and identity to determine whether equation (1) holds. If not, the auditing request is rejected; otherwise the TPA selects a random c-elements subset I of the set $[1, l]$, and for each $i \in I$ selects a random binary string $(h_1, h_2, \cdots, h_c) \in \{0,1\}^c$. The challenge message

$chal = (i, \{h_i\}_{i \in I})$ locates the subfiles which need to be verified. Finally, the TPA forwards the attribute set S_u and $chal = (i, \{h_i\}_{i \in I})$ to the CSP.

ProofGen: Once receiving $chal = (i, \{h_i\}_{i \in I})$ from the TPA, the CSP completes the following steps.

(a) Compute $\sigma = \sum_{i \in I} h_i \sigma_i$, $\mu = \sum_{i \in I} h_i ff_i$.

(b) Output proof information $Proof = \{\sigma, \mu\}$.

ProofVerify: Once receiving $Proof = \{\sigma, \mu\}$, the TPA selects θ random matrices $\boldsymbol{Z}_j \in \mathbb{Z}_q^{n \times m_2}$ of the form $L = \{l_{i,j}\}_{i \in [l], j \in [1+\theta]}$ according to the coefficient matrix in the access policy (L, ρ) , and constructs the verification matrix $\boldsymbol{M}$,where $\boldsymbol{Q}$ and $\boldsymbol{A}_i, i = 1, 2, \cdots \theta$ is public key.

$$\mathrm{M} = \left[\begin{array}{c|cccccc} l_{1,0}\mathrm{Q} & l_{1,1}\mathrm{Z}_1 + l_{1,0}\mathrm{A}_1 & l_{1,2}\mathrm{Z}_2 + l_{1,0}\mathrm{A}_2 & \cdots & \cdots & l_{1,\theta}\mathrm{Z}_\theta + l_{1,0}\mathrm{A}_\theta \\ l_{2,0}\mathrm{Q} & l_{2,1}\mathrm{Z}_1 + l_{2,0}\mathrm{A}_1 & l_{2,2}\mathrm{Z}_2 + l_{2,0}\mathrm{A}_2 & \cdots & \cdots & l_{2,\theta}\mathrm{Z}_\theta + l_{2,0}\mathrm{A}_\theta \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{l,0}\mathrm{Q} & l_{l,1}\mathrm{Z}_1 + l_{l,0}\mathrm{A}_1 & l_{l.2}\mathrm{Z}_2 + l_{l,0}\mathrm{A}_2 & \cdots & \cdots & l_{l,\theta}\mathrm{Z}_\theta + l_{l,0}\mathrm{A}_\theta \end{array}\right] mod\ q$$

(a) If the set of attributes $S_u \in U$ satisfies the access policy (L, ρ) , then there must exist a set of coefficients $\tilde{g} = \{\tilde{g}_{\rho(1)}, \tilde{g}_{\rho(2)}, \cdots, \tilde{g}_{\rho(l)}\} \in \mathbb{Z}^l$ such that $[\tilde{g}_{\rho(1)}, \tilde{g}_{\rho(2)}, \cdots, \tilde{g}_{\rho(l)}] \cdot L = [1, 0, \cdots, 0]$ holds.

(b) The TPA constructs the expansion matrix $\boldsymbol{T}_g$ of the unit matrix $\boldsymbol{I}_n$ using the coefficients $g \in \mathbb{Z}^l$ such that $\boldsymbol{T}_g = [g_{\rho(1)}\boldsymbol{I}_n \mid g_{\rho(2)}\boldsymbol{I}_n \mid \cdots \mid g_{\rho(l)}\boldsymbol{I}_n]$ holds.

(c) Meanwhile, let $\boldsymbol{Z}_{\theta+1}, \cdots, \boldsymbol{Z}_l$ be the zero matrix and the expansion matrix be $\tilde{\boldsymbol{M}}$

$$\widetilde{\mathrm{M}} = \left[\begin{array}{c|cccccc} l_{1,0}\mathrm{Q} & l_{1,1}\mathrm{Z}_1 + l_{1,0}\mathrm{A}_1 & l_{1,2}\mathrm{Z}_2 + l_{1,0}\mathrm{A}_2 & \cdots & l_{1,\theta}\mathrm{Z}_\theta + l_{1,0}\mathrm{A}_\theta & \cdots & l_{1,l}\mathrm{Z}_l + l_{1,0}\mathrm{A}_l \\ l_{2,0}\mathrm{Q} & l_{2,1}\mathrm{Z}_1 + l_{2,0}\mathrm{A}_1 & l_{2,2}\mathrm{Z}_2 + l_{2,0}\mathrm{A}_2 & \cdots & l_{2,\theta}\mathrm{Z}_\theta + l_{2,0}\mathrm{A}_\theta & \cdots & l_{2,l}\mathrm{Z}_l + l_{2,0}\mathrm{A}_l \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{l,0}\mathrm{Q} & l_{l,1}\mathrm{Z}_1 + l_{l,0}\mathrm{A}_1 & l_{l.2}\mathrm{Z}_2 + l_{l,0}\mathrm{A}_2 & \cdots & l_{l,\theta}\mathrm{Z}_\theta + l_{l,0}\mathrm{A}_\theta & \cdots & l_{l,l}\mathrm{Z}_l + l_{l,0}\mathrm{A}_l \end{array}\right] mod\ q$$

(d) The TPA calculates $v_j = H_2(ID \| name \| j)$, makes $\sum_{i \in I} \boldsymbol{v}_i = \boldsymbol{v}$, and checks whether the equation (3)(4) holds. If the verification is valid then return “1”, otherwise the verification fails back to “0”.

$$(\mathrm{T}_g \cdot \widetilde{\mathrm{M}}) \cdot \sigma = \mu + \mathrm{v}\, mod\ q \tag{3}$$

$$0 < \|\psi\| \le sl\sqrt{2m_1 + km_2} \tag{4}$$

Revoke: According to the revocation list and the user attribute set, the AC performs the following steps.

(a) When the user resigns, the AC revokes the user identity and adds the revoked user identity $\{ID_1, \cdots, ID_p\}$ to the revocation list. Suppose the user with identity ID', whose identity is revoked, uploads data to the private cloud, and the private cloud verifies that equation (1) does not hold, thus user $ID^{'}$ can no longer upload or read data to the private cloud, and the updated revocation list is $RL^{'}$.

(b) When the user position is transferred, the attributes corresponding to the user are changed and the AC is reauthorized. The changed user attribute set is recorded as $\boldsymbol{B}_u$, and the user identity remains unchanged. When the user sends an audit request to the auditor for the data originally uploaded to the private cloud, the auditor calculates $H_2(\boldsymbol{B}_u)$ and $H_1(ID)$ based on the user attribute set and the user identity, then equation (6) holds and the auditor rejects the auditing request. Thus, the user whose attributes have changed can no longer read and verify the integrity of the data previously uploaded to the private cloud. The system model after user attribute change is shown in Figure 2.

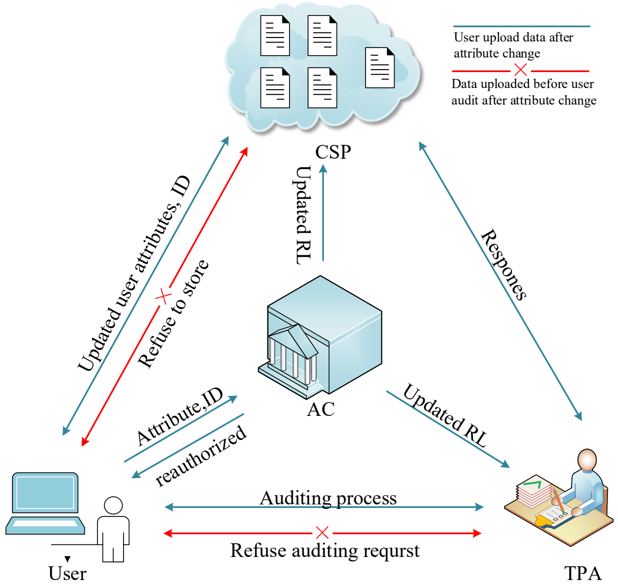
User upload data after attribute change
Data uploaded before user audit after attribute change
CSP
Updated RL
Updated user attributes, ID
Refuse to store
Respones
AC
Attribute,ID
reauthorized
Updated RL
Auditing process
Refuse auditing requrst
User
TPA

Fig. 2. User attributes occurring change system model [Owner-draw]

5 Correctness and security analysis

5.1 Correctness proof

$$
\begin{aligned}
A_u \cdot \sigma &= A_u \sum_{i\in I} h_i \sigma_i \\
&= \sum_{i\in I} h_i \boldsymbol{\beta} \bmod q \\
&= \sum_{i\in I} h_i (ff_i + v_i) \bmod q \\
&= \sum_{i\in I} (h_i ff_i + h_i v_i) \bmod q \\
&= \mu + \boldsymbol{v} \bmod q\,.
\end{aligned}
$$

Therefore, equation (3) holds and there is $0 < \|\sigma_i\| \leq s\sqrt{2m_1 + km_2}$, and thus equation (4) also holds.

5.2 Security proof

In this section, we prove the safety of the proposed scheme by using the following theorem.

Theorem1: If the SIS problem is hard, then the scheme is unforgeable under chosen attribute set and chosen message attacks.

Proof. If there exists an adversary $\mathcal{A}$ who succeeds in forging a signature with probability advantage ε , then the adversary $\mathcal{A}$ can solve $SIS_{n,q}$ problem with non-negligible probability using challenger $\mathcal{C}$'s algorithm. If $\mathcal{C}$ is given the SIS problem instance construction matrix $\boldsymbol{A}_u^*$, with the help of $\mathcal{A}$ finding a nonzero vector σ^* such that $\boldsymbol{A}_u^* \cdot \sigma^* = y \bmod q$ and $\| \sigma^* \| \leq \beta$, the adversary $\mathcal{A}$ and challenger $\mathcal{C}$ interact as follows.

At the beginning of the game, the adversary $\mathcal{A}$ determines the challenge attribute set $\tilde{S}^*$, signature policy (L^*, ρ^*) and file f^*. The challenger $\mathcal{C}$ maintains two lists L_1 , L_2 , which are initially empty. According to the system security parameters n , the challenger $\mathcal{C}$ simulates the generation of public parameters, $pp = \{\boldsymbol{A}^*, \{\boldsymbol{A}_i^*\}_{i \in Att}, \{\boldsymbol{C}_i^*\}_{i \in [1,l]}, s, H_1, H_2, H_3\}$.

Extract Query : The adversary $\mathcal{A}$ selects a user attribute set $\tilde{S}'$, and asks the challenger $\mathcal{C}$ for the private key, but requires $\tilde{S}' \neq \tilde{S}^*$. If user identity $ID^* \neq ID$, the challenger $\mathcal{C}$ sets $H(ID^*) = \boldsymbol{R}^*$ and returns $\mathbf{T}_{ID^*} = \perp$ to the adversary $\mathcal{A}$. Otherwise, the challenger $\mathcal{C}$ runs the *RandBasis* algorithm to generate $\mathbf{T}_{ID^*}$ for the adversary $\mathcal{A}$, and uses L_1 to save the query result.

(a) Let the set of user attributes be $U^* = \{U_{att_1}, \cdots, U_{att_{k^*}}\}$. If $U_{att_i}^* = Att_i$, then let $U_{att_i}^* = \boldsymbol{A}_i^*$, otherwise $U_{att_i}^* = 0$. Obtain the attribute matrix as $\boldsymbol{A}_{u^*} = (\boldsymbol{Q}^* \mid \boldsymbol{A}_1^* \mid \boldsymbol{A}_2^* \mid \cdots \mid \boldsymbol{A}_{k^*}^*) \in \mathbb{Z}_q^{n \times (2m_1 + k^* m_2)}$.

(b) The adversary $\mathcal{A}$ performs hash query and calculates $ff_j^* = H_4(C_j^{(f_j^*)})$, $\boldsymbol{\beta}^* = ff_j^* + \boldsymbol{\gamma}_j$ and lets $A_{ff_j} = [\boldsymbol{A}_u \| \boldsymbol{C}_j^{(f_j)}]$, $\boldsymbol{\gamma}_j = H_3(ID^* \| name^* \| j) \in \mathbb{Z}_q^n$, $\boldsymbol{A}_{ff_j}^* = [\boldsymbol{A}_u^* \| \boldsymbol{C}_j^{(f_j)}]$.

(c) The challenger $\mathcal{C}$ runs the steps of the key generation phase to generate the private key SK_{ID}^* and public key PK_{ID}^* .

(d) The challenger $\mathcal{C}$ sends the query result to the adversary $\mathcal{A}$, and saves the query result in the list L_1 .

SigGen Query : The adversary $\mathcal{A}$ selects any signature attribute set $\tilde{S}^* \in Att$ and the file $f^* = (f_1, f_2, \cdots, f_l)$. The challenger $\mathcal{C}$ chooses an access policy (L^*, ρ^*), but requires that the challenge attribute set does not satisfy the signature policy $(L^*, \rho^*) \neq \tilde{S}^*$, the adversary $\mathcal{A}$ issues a signature generation query to the challenger $\mathcal{C}$, and the challenger $\mathcal{C}$ determines whether $\tilde{S}^*$ satisfies the access policy to query whether f^* has been queried. If it is not satisfied, the challenger $\mathcal{C}$ looks up the list L_2 and finds the signature σ_i^* corresponding to the message. If satisfied, the challenger $\mathcal{C}$ simulates the above label generation algorithm to generate a signature of f^*.

Return the signature $\sigma^* = \{\sigma_i^*\}_{1 \le i \le l}$ of the attribute subset $\tilde{S}^* \in Att$ on the signature policy (L^*, ρ^*) to the adversary $\mathcal{A}$, and use the list L_2 to save the query result.

Forge : The adversary $\mathcal{A}$ forged audit evidence $\sigma^* = \sum_{i \in I, i \neq k} h_i \sigma_i + h_k \sigma_k^*$, $\mu^* = \sum_{i \in I, i \neq k} h_i ff_i + h_k ff_k^*$.

(a) Assuming that the auditing proof is forged successfully, that is, $proof = (\sigma^*, \mu^*)$ passes the auditing verification, then there are $\sum_{i \in I} \gamma_i = \gamma$, $0 < \| \sigma^* \| \le s\sqrt{2m_1 + k^* m_2}$ and $(\boldsymbol{T}_g \cdot \tilde{\boldsymbol{M}}^*) \cdot \sigma^* = \mu^* + \gamma \bmod q$.

(b) Construct the extended matrix $\boldsymbol{T}_g$ of the identity matrix $\boldsymbol{I}_n$ with the coefficients $g \in \mathbb{Z}^l$,

$\boldsymbol{T}_g = [g_{\rho(1)} \boldsymbol{I}_n \mid g_{\rho(2)} \boldsymbol{I}_n \mid \cdots \mid g_{\rho(l)} \boldsymbol{I}_n]$. Construct the matrix,

$$\widetilde{\mathrm{M}}^* = \left[\begin{array}{c|cccccc} l_{1,0}\mathrm{Q}^* & l_{1,1}\mathrm{Z}_1 + l_{1,0}\mathrm{A}_1^* l_{1,2}\mathrm{Z}_2 + l_{1,0}\mathrm{A}_2^* & \cdots & l_{1,\theta}\mathrm{Z}_\theta + l_{1,0}\mathrm{A}_\theta^* & \cdots & l_{1,l}\mathrm{Z}_l + l_{1,0}\mathrm{A}_l^* \\ l_{2,0}\mathrm{Q}^* & l_{2,1}\mathrm{Z}_1 + l_{2,0}\mathrm{A}_1^* l_{2,2}\mathrm{Z}_2 + l_{2,0}\mathrm{A}_2^* & \cdots & l_{2,\theta}\mathrm{Z}_\theta + l_{2,0}\mathrm{A}_\theta^* & \cdots & l_{2,l}\mathrm{Z}_l + l_{2,0}\mathrm{A}_l^* \\ \vdots & \vdots \quad \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{l,0}\mathrm{Q}^* & l_{l,1}\mathrm{Z}_1 + l_{l,\theta}\mathrm{A}_1^* \; l_{l.2}\mathrm{Z}_2 + l_{l,0}\mathrm{A}_2^* & \cdots & l_{l,\theta}\mathrm{Z}_\theta + l_{l,0}\mathrm{A}_\theta^* & \cdots & l_{l,l}\mathrm{Z}_l + l_{l,0}\mathrm{A}_l^* \end{array}\right] mod\, q$$

Then there is $\boldsymbol{T}_g \cdot \tilde{\boldsymbol{M}}^* = \boldsymbol{A}_u^*$, that is $(\boldsymbol{T}_g \cdot \tilde{\boldsymbol{M}}^*) \cdot \sigma^* = \mu^* + \gamma$, $(\boldsymbol{T}_g \cdot \tilde{\boldsymbol{M}}^*) \cdot \sigma^* = \boldsymbol{A}_u^* (\sum_{i \in I, i \neq k} h_i \sigma_i + h_k \sigma_k^*)$, thus,

$$\mathrm{A}_u^* h_k \sigma_k^* = \mathrm{A}_u^* \sigma^* - \mathrm{A}_u^* \textstyle\sum_{i \in I, i \neq k} h_i \, \sigma_i = \mu^* - \mu = \textstyle\sum_{i \in I, i \neq k} h_i f f_i + h_k f f_k^* - \textstyle\sum_{i \in I,} h_i f f_i = h_k f f_k^*.$$

And equation $\boldsymbol{A}_u^* \sigma_k^* = ff_k^* = \boldsymbol{C}_k^{(f_k^*)} H_2(name \| k)$ holds.

(c) Let $\boldsymbol{y} = \boldsymbol{C}_k^{(f_k^*)} H_2(name \| k)$, then the forged signature of ff_k^* can be calculated to satisfy $\boldsymbol{A}_u^* \sigma^* \bmod q = \boldsymbol{y}$.

Therefore, the *SIS* problem is solved by the above interactive adversary $\mathcal{A}$ with non-negligible advantage ε.

6 Performance analysis

6.1 Functionality comparison

In this section, a comparison is presented between our proposed scheme with the same type of data integrity auditing schemes [8], [10], [13], [14], [15]in terms of whether the authorization center is trusted, fine-grained access control, quantum resistance, public auditing and user revocation, and the results are shown in Table 1.

Both our proposed scheme and the above schemes have realized the public auditing of outsourced data. Schemes [14], [15] are identity-based data integrity auditing with key escrow problem. Scheme [8] and scheme [13] are attribute-based data integrity auditing schemes achieving one-to-many authorization, but they cannot resist quantum attacks. Scheme [10] provides a certificateless data integrity audit scheme to solve the key escrow problem but it does not achieve fine-grained access control of user.

Compared with the above schemes, our proposed scheme uses lattice randomization algorithm to prevent authorization centers from forging signatures and protect enterprise data security, thus there is no need for a full trusted authorization center. Our proposed scheme not only realizes fine-grained control of user permissions and dynamic management of user, but also resists quantum attacks. Therefore, our proposed scheme achieves more comprehensive functions and has stronger security and practicality.

Table 1. Functionality comparison [Owner-draw]

Schemes	Authorization center	Fine-grained	Post-quantum security	Public auditing	Revocable
8	Trusted	√	×	√	×
10	Trusted	×	√	√	×
13	Trusted	×	×	√	√
14	Trusted	×	√	√	×
15	Trusted	×	×	√	×
Ours	Untrusty	√	√	√	√

6.2 Performance comparison

Table 2. Symbol and their meaning [Owner-draw]

Symbol	Meaning
c	The number of the challenged block
l	The total number of data blocks
T_{Exp}	The time cost of exponentiation on G_1
T_{mul}	The time cost of performing once multiplication on vectors, G_1
T_{pair}	The time cost of bilinear pairings on G_1
T_{Hash}	The time cost of performing once hash function
T_{Add}	The time cost of performing once addition on G_1
T_{SAM}	The time cost of performing once *SamplePre* algorithm

In this section, we make a comparison of the computational cost between our proposed scheme with the existing schemes [10] and [16] in Table 3. For the convenience of comparison, we summarize the parameters used and their meanings, as shown in Table 2. In the data auditing phase, the computational cost of the auditor in our scheme is $T_{Hash}+(cn+nm_1+nm_2+n)T_{mul}$, and the computational cost of the auditor in scheme 16 is $3T_{pair}+(2c+1)T_{Exp}+(2c-1)T_{Add}+(c+1)T_{Hash}$, which requires the execution of bilinear pairs and modulo exponential operations, while our scheme does not require the execution of the above operations and can resist to quantum attacks with higher security. In scheme 10, the computational overhead of the TPA is $(n+c+1)T_{Hash}+(4mn+n^2+nc+n+2m)T_{mul}+T_{SAM}$ and more $Hash$ operations need to be executed, and the auditor is also involved in the proof generation process, thus the auditor also needs to execute $SamplePre$ algorithms, while our scheme only needs to execute one operation in the data validation process.

Compared with the scheme [10], [16], our scheme reduces the computational burden of the auditor and reduces the computational cost of the auditor. Therefore, it alleviates the efficiency bottleneck problem that occurs on the TPA side.

Table 3. Computational Cost Comparison [Owner-draw]

Schemes	Data Auditing Phase
10	$(n+c+1)T_{Hash}+(4mn+n^2+nc+n+2m)T_{mul}+T_{SAM}$
16	$3T_{pair}+(2c+1)T_{Exp}+(2c-1)T_{Add}+(c+1)T_{Hash}$
Ours	$T_{Hash}+(cn+nm_1+nm_2+n)T_{mul}$

7 Conclusions

To address the management chaos in enterprise private cloud data sharing, we propose a revocable attribute-based data integrity auditing scheme on lattice. In the key extraction phase, user attribute sets are embedded to achieve fine-grained management of user permissions. The key escrow problem is solved by the lattice-based randomization algorithm, which effectively prevents authorized center from forging signatures by using private keys and has stronger security. Moreover, our scheme realizes dynamic management of users to meet the requirements of flexible replacement of data managers when enterprises use private clouds. Under the hardness assumption of SIS problem, it is proved that our scheme can resist quantum attacks and is against selective attribute set attack. The analysis results show that the proposed scheme has the advantages of efficiency and practicability.

Acknowledgements

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

References

1. N. Doukas, O. Markovskyi, N, Bardis, Hash function design for cloud storage data auditing [J]. Theoretical Computer Science, 2019, 800: 42-51.
2. J. Tian, J. Xuan, Cloud data integrity verification scheme for associated tags[J]. Computers & Security,2020,95: 101847.
3. J. Li, H. Yan, Y. Zhang, Identity-based privacy preserving remote data integrity checking for cloud storage[J]. IEEE Systems Journal, 2020, 15(1): 577-585.
4. M. Tian, C. Gao, J. Chen, Identity-based cloud storage integrity checking from lattices[J]. Journal on Communications, 2019, 40(4): 128-139.
5. G. Ateniese, R. Burns, R. Curtmola, et al. Provable data possession at untrusted stores[C]//Proceedings of the 14th ACM conference on Computer and communications security. 2007: 598-609.
6. A. Juels, JR. Kaliski, PORs: Proofs of retrievability for large files[C]//Proceedings of the 14th ACM conference on Computer and communications security. 2007: 584-597.
7. X. Zhang, C. Huang, Y. Zhang, et al. LDVAS: Lattice-based designated verifier auditing scheme for electronic medical data in cloud-assisted WBANs[J]. IEEE Access, 2020, 8: 54402-54414.
8. Y. Yu, Y. Li, B. Yang, et al. Attribute-based cloud data integrity auditing for secure outsourced storage[J]. IEEE Transactions on Emerging Topics in Computing, 2017, 8(2): 377-390.
9. Y. Wang, C. Chen, Z. Chen, et al. Attribute-Based User Revocable Data Integrity Audit for Internet-of-Things Devices in Cloud Storage[J]. Security and Communication Networks, 2020, 2020.
10. H. Li, Y. Wang, X. Fu, et al. PSCPAC: Post-quantum secure certificateless public auditing scheme in cloud storage[J]. Journal of Information Security and Applications, 2021, 61: 102927.
11. X. Liu, Y. Luo, X. Yang, et al. Lattice-Based Proxy-Oriented Public Auditing Scheme for Electronic Health Record in Cloud-Assisted WBANs[J]. IEEE Systems Journal, 2022, 16(2): 2968-2978.
12. D. Micciancio, O. Regev, Worst-case to average-case reductions based on Gaussian measures[J]. SIAM Journal on Computing, 2007, 37(1): 267-302.
13. J. Gudeme, S. Pasupuleti, R. Kandukuri, Attribute-based public integrity auditing for shared data with efficient user revocation in cloud storage[J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(2): 2019-2032.
14. Z. Liu, Y. Liao, X. Yang, et al. Identity-based remote data integrity checking of cloud storage from lattices[C]//2017 3rd International Conference on Big Data Computing and Communications (BIGCOM). IEEE, 2017: 128-135.
15. Y. Yu, M. Au, G. Ateniese, et al. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2016, 12(4): 767-778.

16. B. Wang, B. Li, H. Li, et al. Certificateless public auditing for data integrity in the cloud[C]// 2013 IEEE conference on communications and network security (CNS). IEEE, 2013: 136-144.