Hindawi
Security and Communication Networks
Volume 2019, Article ID 1354346, 11 pages
https://doi.org/10.1155/2019/1354346

Research Article

Integrity Audit of Shared Cloud Data with Identity Tracking

Yun Xue Yan,[1,2] Lei Wu ,[1,2,3] Wen Yu Xu,[1,2] Hao Wang,[1,2] and Zhao Man Liu[1,2]

[1]*School of Information Science and Engineering, Shandong Normal University, Jinan, China*
[2]*Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, Jinan, China*
[3]*Shandong Provincial Key Laboratory of Software Engineering, Jinan, China*

Correspondence should be addressed to Lei Wu; wulei@sdnu.edu.cn

Received 5 January 2019; Accepted 19 February 2019; Published 6 March 2019

Guest Editor: Pelin Angin

More and more users are uploading their data to the cloud without storing any copies locally. Under the premise that cloud users cannot fully trust cloud service providers, how to ensure the integrity of users' shared data in the cloud storage environment is one of the current research hotspots. In this paper, we propose a secure and effective data sharing scheme for dynamic user groups. (1) In order to realize the user identity tracking and the addition and deletion of dynamic group users, we add a new role called Rights Distribution Center (RDC) in our scheme. (2) To protect the privacy of user identity, when performing third party audit to verify data integrity, it is not possible to determine which user is a specific user. Therefore, the fairness of the audit can be promoted. (3) Define a new integrity audit model for shared cloud data. In this scheme, the user sends the encrypted data to the cloud and the data tag to the Rights Distribution Center (RDC) by using data blindness technology. Finally, we prove the security of the scheme through provable security theory. In addition, the experimental data shows that our proposed scheme is more efficient and scalable than the state-of-the-art solution.

1. Introduction

As an emerging network storage technology, cloud storage has been extended and developed in cloud computing. Cloud computing systems are transformed into cloud storage systems when the core of computing and processing is to store and manage massive data. In simple terms, cloud storage is an emerging solution that puts storage resources on the cloud for people access.

The user can access data on the cloud easily through any connected device whenever and wherever. Through data storage and sharing services in cloud computing, group members can share data in the form of a group. As a member of a group, users can not only access the shared data, but also modify the shared data. While cloud computing makes it easier for users to share data, users are still concerned about the security of data, especially the integrity of data, due to some security factors in cloud storage. The effective way is to use third party auditor (TPA) to achieve the purpose of validating shared data integrity. However, third party auditor (TPA) can obtain the block identifier (that is, the identity of each shared block signer) during the process of verifying the data integrity. If these identity information and confidential information in the shared data group cannot get effective protection, they will be leaked to a third party auditor (TPA) such as the situations that user in the group plays a crucial role or data block in the shared data has higher value.

Although the current public auditing scheme for sharing data solves the problem of user identity protection, it also brings dynamic changes in the group. However, the identity of group members who maliciously modify the shared data cannot get traced. We can observe that the amount of computation is comparatively large during the signature of data blocks by cloud users, which takes users a long time with limited resources. This paper proposes an auditing scheme that supports user identity tracking and lightweight sharing of cloud data, which enables traceability of user identities and reduces the burden on the resource constrained users. Using the data storage and sharing services provided by cloud server, legitimate users can easily form a group by sharing data with each other. That is to say, the users can create data and share it with others in the group. Users in the group can not only access the shared data, but also modify the shared data. Although cloud service providers

provide users with a secure and reliable storage environment as much as possible, data integrity can still be compromised. For example, it is considered that operational errors, data hardware and software failures, may lead to data tampering and data loss. This series of problems happened to us [1].

2. Related Work

Users always pay more attention to data security in the cloud. In recent years, data integrity schemes have become one of the research hotspots. With the help of data integrity schemes, any data corruption or deletion can be discovered in time and then necessary measures can be taken to recover the data. To develop a better understanding of data integrity schemes, we carry out the relevant work from the audit model, soundness, and other aspects.

Performance. Many researchers have proposed a series of schemes to this problem. On the one hand, how to solve the problem of user revocation? Wang et al. [2–7] noticed the problem of shared data integrity verification and proposed a public auditing method that supports efficient user revocation for shared data. To sum up, this scheme introduces proxy resignature technology to solve the problem. However, when the user is revoked, the cloud server is allowed to replace the previously signed data block of the revoked user to a legal group instead of the group member, which can cause efficiency problem. In addition, in scheme [8], the authors propose to enable efficient user revocation in identity-based cloud storage auditing for shared big data. On the other hand, Yu et al. [8, 9] proposed the issue of key security among cloud users. In these schemes, the key exposure in one time period does not affect the security of cloud storage auditing in other time periods and verifiable out-sourcing of key updates.

Identity Privacy. With the development of related technologies in cloud computing, public audit of shared data integrity has attracted more and more attention. Yu et al. [10] proposed that the storage and sharing services of cloud servers allow users to share data in the form of a group. As a group member, they have the right to view and modify shared data. Although users can easily share data, data integrity issues remain [11, 12]. Using TPA for public auditing results in the leakage of user's identity privacy [13]. Wang et al. [14] fully considered the confidentiality of the data in the public audit process and proposed a privacy scheme that used ring signature to protect group member. Adopting the ring signature can ensure that the TPA protects the user's identity privacy while verifying the integrity of the data. However, the efficiency of the scheme is reduced by the increasing number of team members. Meanwhile, the client also takes a lot of computing. Therefore, the scheme does not apply to large user groups. Shen et al. [15] proposed a lightweight auditing scheme for shared data privacy protection, taking full account of the computational limitations of the resource constrained client. Using data blindness methods, the scheme allows (TPM) Third Party Medium instead of group users to sign the data. It not only reduces the burden on the client, but also ensures the privacy of identity during public auditing. Thus the identity of the data owner can be protected. However, this scheme does not support group dynamics and the traceability of data blocks. Wang et al. [16] proposed another public audit method for sharing data privacy protection. Using dynamic broadcast technology, group members can be signed as the owner of the data when modifying the shared data, thereby protecting the privacy of the group members. It not only realizes the dynamic operation of data by group members, but also supports group dynamics. However, this scheme does not protect the identity of data owner, making the TPA steal the identity of the data owner during public auditing, and it does not support the traceability of data blocks.

Public Auditability/Private Auditability. The first method [17] allows only the data owner to audit. The second [18] method allows a third party auditor to audit. The audit process in both approaches is performed without retrieving the remote data. If only the data owner can verify the integrity of the outsourced data, then this scheme is considered to provide private auditability. However, in some cases, it is not practically feasible for the data owner to remain online all time for data integrity verification. Hence, the data owner can delegate this responsibility for integrity verification to a third party auditor or other users. A data integrity scheme must have public auditability property to support this audit delegation.

Dynamic Data Handling. Data can be either static (backup or archival data) or dynamic nature (supporting operations like insertion, deletion, and modification). Providing integrity for dynamic data is more challenging than static data or just attaching data. Most of the schemes proposed in the literature are not able to handle dynamic data, such as the description of the schemes [19, 20] dynamic data handling characteristic demands that data integrity should remain intact, even after insertion, deletion, or modification.

Soundness. An untrusted server cannot able to deceive a challenge request. In the schemes of Wang [21] and Zhang et al. [22], the soundness property of data integrity schemes ensures data reliability. Data integrity schemes are designed to prevent tampering. Therefore, if metadata is tampered with or corrupted intentionally or unintentionally by the CSP, this should be timely identified by a data integrity scheme. If the CSP can pass a challenge request without holding the data or with corrupted data, then a client will never be able to identify data corruption promptly, and the value of the data will be lost. Therefore, a good data integrity scheme requires that the server's response must be reliable.

Privacy Preserving. Privacy protection should be emphasized in the process of data integrity verification. As involved in the scheme [23], privacy concerns are introduced due to public verifiability. On the premise that the data owner will not allow the disclosure of his private data to a third party auditor, the privacy preservation property demands that a third party auditor should not obtain any confidential information about the user's data but can still verify the integrity of outsourced data.

Fairness. In the scheme [24], fairness means that a data integrity scheme should provide protection for an honest CSP against legitimate but dishonest users, who may attempt to accuse CSP of manipulating the outsourced data. If a data integrity scheme does not support fairness, it means dishonest users can damage CSP reputation.

Organization. The organization of the paper is as follows: the first part introduces the research status and background of cloud sharing data; the second part introduces the relevant work; the third part introduces the relevant knowledge; the fourth part describes the system model and each function of its entity, and describes the integrity audit scheme in detail; the fifth part analyses the security of the scheme, including the correctness analysis, unforgeability analysis, and proof of identity privacy by using provable security theory; the sixth part analyses the performance of the proposed scheme, including the functional comparison and efficiency analysis among different schemes. Finally, according to the advantages and disadvantages of this paper, we will formulate our next research direction.

3. Preliminaries

3.1. Bilinear Pairings. Let G_1 and G_2 be two multiplicative groups of the prime order q, and g_1, g_2 be generators of group G_1. A bilinear pairing is a map $\hat{e}$: $G_1 \times G_1 \longrightarrow G_2$ with following properties.

(1) Bilinearity

$$\begin{aligned} &\text{For } \forall g_1, g_2 \in G_1 \text{ and } a, b \in_R Z_q^*, \\ &\text{there is } \hat{e}\left(g_1^a, g_2^b\right) = \hat{e}\left(g_1, g_2\right)^{ab} \end{aligned} \tag{1}$$

(2) Nondegeneracy

$$\hat{e}\left(g_1, g_2\right) \neq 1 \tag{2}$$

(3) Computability. There is an efficient algorithm to compute this pairing.

3.2. Data Blindness. In general, the blindness of the data is that user A passes the encrypted data to user B and user B cannot infer the plaintext of user A based on these data. Therefore, users are protected as privacy. Among them, a simpler and less computational scheme is proposed in the paper, which can complete the blinding of data. The method is as follows: user A blinds the data block by using the random function and sends it to user B. User B cannot obtain the original data.

3.3. Security Theory Assumption

Definition 1 (DL problem). Unknown $a \xleftarrow{R} z_q^*$, g is the generator. Given g^a calculate α.

Definition 2 (DL assumption). The probabilistic advantage of algorithm B to solve the DL problem in probabilistic polynomial time is

$$Adv_{DL}(B) = \text{pr}\left[a \longleftarrow B\left(g, g^a\right)\right] \tag{3}$$

If $Adv_{DL}(B)$ is negligible, it is called the DL problem which is difficult.

Definition 3 (DCDH problem). Known $a, b \xleftarrow{R} z_P^*$, given $g^{1/a}$ and g^b, calculate g^{ab}.

Definition 4 (DCDH assumption). The probability that algorithm B solves the DCDH problem in probabilistic polynomial time is

$$Adv_{DCDH}(B) = \text{pr}\left[g^b \longleftarrow B\left(g, g^{ab}, g^a\right)\right. \tag{4}$$

If $Adv_{DCDH}(B)$ is ignored, it is difficult to call the DCDH problem.

3.4. Dynamic Broadcast Technology. Broadcast encryption technology is capable of transmitting encrypted information to group members over a broadcast channel. During the dissemination of this information, only members of the group can decrypt the message. Compared with traditional BE, BE can effectively support the dynamic changes of the group.

3.5. Data Sharing Integrity Verification Threat Target

Cloud Server Storage Problem. Cloud servers face the problems in situations where data is lost or data preservation is incomplete. Considering the interest of cloud service providers, to protect their reputation, they may have the potential to defraud public auditors.

Data Leakage Problem. In the process of integrity auditing performed by the third party audit, when the cloud service provider submits the certificate to the TPA for complete public verification, the cloud service provider also sends the linear combination value of the data to the third party audit. This leads to the possibility that third parties may steal content from shared data and infer the identity of the relevant user.

Data Tamper Problem. As for shared data in the cloud, team members may make malicious changes, resulting in the fact that shared data is not available. However, due to the fact that users cannot be traced back to a particular cloud, resulting in data being tampered with, so they still cannot determine the user's identity.

4. Our Construction

4.1. System Architecture

Rights Distribution Center (RDC). Figure 1 shows the cloud shared data model. In the process of data integrity verification, users, third party audit, and cloud service provider are often involved in privacy disclosure and user identity

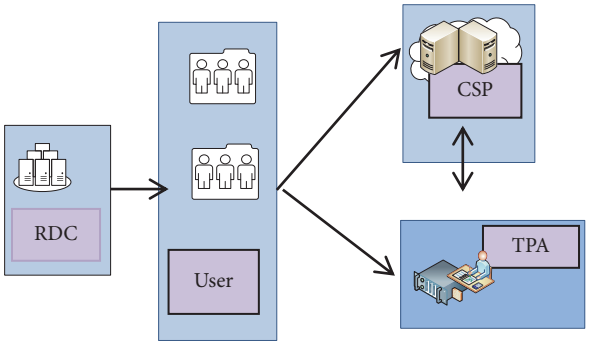

Figure 1: Cloud sharing data model.

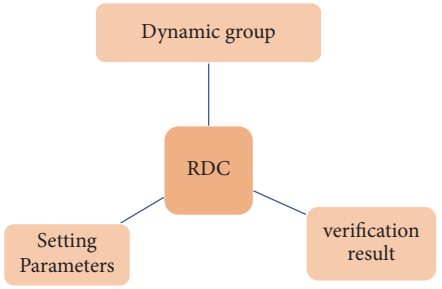

Figure 2: Right Distribution Center.

traceability issues. In this paper, by introducing the Rights Distribution Center, as shown in Figure 2 the users will be reasonably grouped and the RDC will record the operations of the data performed by the user. The RDC first performs an initialization operation to set global parameters $(G_1, G_2, \hat{e}, g, \mu, PK)$ for the system. RDC selects x as its own private key and $X_j \in Z_q^*$ as the private key of the member M_j and sets a hash function $H: Z_q^* \longrightarrow G_1$. Secondly, the RDC generates auxiliary information of the relevant data according to the (id_i, δ_i) sent by the user. The relevant information is counted in the table. Finally, when the user requests to operate on the data, the RDC will record the operation of the corresponding user to achieve identity tracing.

User. As a member of the cloud sharing data service, after registering an account, the user needs to insert, modify, and delete his or her own data. As shown in Figure 3. In the scheme, when the user sends the data to the cloud service provider, the user first performs data blinding operation on the data. On the one hand, the user blinds the data using the pseudorandom function and sends the blinded (5) to the cloud service provider.

$$m_i' = m_i + a_i \tag{5}$$

On the other hand, the user sends the tag δ_i generated by his data block to the Rights Distribution Center. Finally, the cloud user generates its own integrity verification request.

$$\sigma_i = \sigma_i' \bullet \left(\frac{PK}{g^{x_j}}\right)^{-r_j} \left(H(id_i) \bullet \mu^{\delta_i}\right)^{x_j} \tag{6}$$

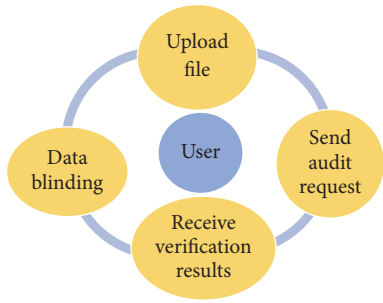

Figure 3: User.

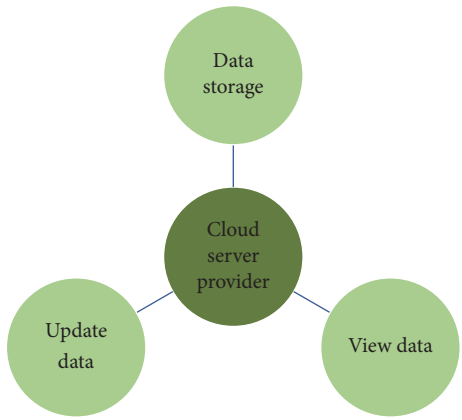

Figure 4: Cloud server provider.

According to the auxiliary information sent by RDC and its own private key, the audit request is sent to the TPA. The third party audit center verifies the integrity of the data and returns the results to the user.

Cloud Service Provider (CSP). The cloud storage service provides data owners with data storage capabilities, so that the client does not need to back up locally when using it, reducing the pressure on local storage. When the cloud service provider receives the challenge of the TPA, the cloud service provider generates evidence to indicate the integrity of the data and sends it to the TPA based on the stored file. According to the proposed scheme, on the one hand, the cloud service provider processes the data sent by the user and obtains the original data through processing. It will use the pseud random key π_k to get the original data m_i and store the data in the next step. On the other hand, according to the challenge sent by the TPA, the cloud service provider calculates the δ_i corresponding to m_i. It calculates the linear combination value u of the sample block, and sends proof = (σ, u) to TPA, from which the TPA detects whether the data is complete. Figure 4 provides a brief description of the cloud service provider.

Third Party Audit (TPA). In Figure 5, when receiving a user's audit request, the TPA first sends a challenge to the cloud service provider and then verifies the data based on the evidence returned by the cloud service provider to determine

Table 1: The meaning of the notation.

Notation	Meaning	Notation	Meaning
$\hat{e}$	A bilinear pairing map	name	Representative data identifier
H	A hash function	id_i	The identity of the user
G_1 G_2	Multiplicative groups with order q	m_i'	The blinded i_{th} block
Z_q^*	A prime field with nonzero elements	v_i	Random select from Z_q^*
F	The data file shared in the cloud	x_j	The group member's secret key
m_i	The i_{th} block of the shared cloud data file,that is $F=\{m_1,m_2,m_3\cdots\cdots m_n\}$	x_j'	The group member's partial secret key
i	Represents data block number	X	The RDC's secret key
chal	The challenge message sent to CSP by the TPA	proof	The proof message sent to TPA by the CSP
audit request	The audit message sent to TPA by the RDC	TPA	Third party audit
RDC	Rights distribution center	User	Receive cloud service
CSP	Cloud service provider	UIT	User identity table

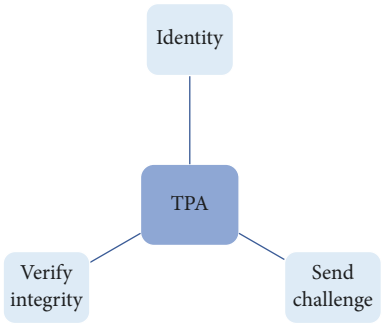

Figure 5: Third party audit.

whether the data is complete. Finally, the TPA returns the result of the integrity verification to the user. If it is complete, it returns 1; else it returns 0. In this scheme, we first initialize the user identity hash value as a reservation to the TPA. It will be used to verify the identity of the user. After the identity of the user is verified by the TPA, the TPA sends a challenge to the cloud service provider. Receiving the evidence returned by the cloud service provider, the TPA verifies whether (7) is true to judge the integrity of the data.

$$\hat{e}\left(\prod_{i\in I} H\left(id_i\right)^{v_i} \bullet u^{\mu}, PK\right) = \hat{e}\left(g,\sigma\right) \tag{7}$$

4.2. The Proposed Scheme. To verify the integrity for shared data efficiently [15, 25–30], our scheme is designed to achieve the following goals.

Cloud Data Privacy. In our scheme, we need to make sure that TPA does not know the real data from the user. At the same time, it cannot get the content of the real data from the cloud response in the audit phase.

Audit Soundness. When the cloud stores the data intact, the cloud server can be validated by TPA.

Identity Privacy. TPA cannot determine which user sent the audit request during the validation of data integrity.

The cloud sharing model mentioned in this paper includes RDC, CSP, TPA, and Client. In the following introduction, the relevant notations are shown in the Table 1. The details of the algorithm are shown in Figure 6.

(1) Setup. User can be expressed as M_j(j=1, 2...s) in the scheme. The initialization work is completed by RDC. RDC generates two multiplicative groups G_1,G_2,

$$\hat{e}\colon G_1 \times G_1 \longrightarrow G_2 \tag{8}$$

RDC selects two independent generators $g, \mu \in G_1$, chooses a hash function $H:Z_q^* \longrightarrow G_1$, and calculates

$$\mathrm{PK} = g^x \tag{9}$$

RDC selects $X_j \in Z_q^*$ as the private key of member M_j and selects x as its own private key. Select r_j and calculate g^{r_j}. So the public parameters are $(G_1, G_2, \hat{e}, g, \mu, \mathrm{PK})$. RDC distributes the private key to user.

(2) Encryption. The user selects the file and divides the file into blocks $M = \{m_1, m_2, m_3 \cdots\cdots m_n\}$. The user's identity can be identified as id_i. For each file block we can make the following operations. First, the file is blinded; we blind the data by using pseudo-random functions. We use $a_i = f_{\pi_K}(i, name)$. Each blinded file block is $m_i' = m_i + a_i$. Second the user generates a file label for each file block by using a short signature Tag_{m_i}. For convenience, we use δ_i to represent Tag_{m_i}. On the one hand, the user sends (m_i', π_k) to the CSP. On the other hand, the user sends (id_i, δ_i) to RDC. Once RDC receives the user's (id_i, δ_i), it will generate user's identity table referred to as UIT, which is shown in Table 2. RDC chooses x as its own private key and calculates

$$x_j' = x - x_j \quad \left(j = 1, 2, \cdots s\right) \tag{10}$$

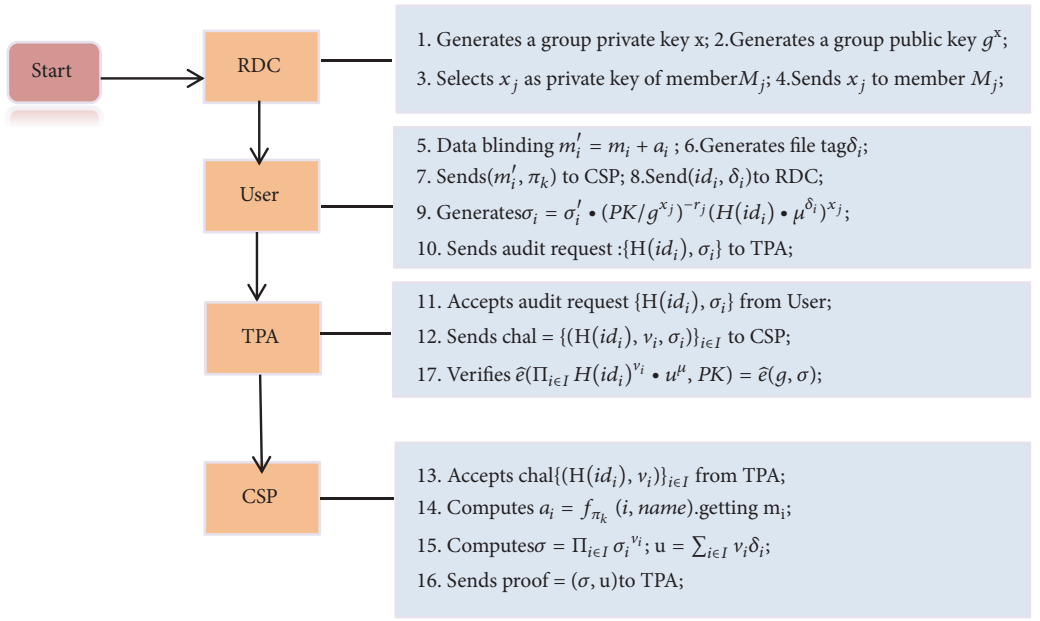

Figure 6: The relationships and interaction order of four entities.

Table 2: User identification table.

No	id	tag	Identity hiding	Member private key
1	id_1	δ_1	H(id_1)	x_1
2	id_2	δ_2	H(id_2)	x_2
3	id_3	δ_3	H(id_3)	x_3
…	…	…	…	…
n	id_n	δ_i	H(id_n)	x_j

Table 3: Stores related user hash values.

No	The hash of the user's identity
1	H(id_1)
2	H(id_2)
3	H(id_3)
…	…
n	H(id_n)

Using δ_i and x_j', RDC calculates σ_i'.

$$\sigma_i' = \left(H(id_i) \bullet \mu^{\delta_i} g^{r_j}\right)^{x_j'} \tag{11}$$

Send σ_i' to the user. The RDC sends the hash value to the TPA. As shown in Table 3, TPA keeps a copy of the legal user's identity table.

(3) Audit Request. User calculates σ_i by σ_i'.

$$\sigma_i = \sigma_i' \bullet \left(\frac{PK}{g^{x_j}}\right)^{-r_j} \left(H(id_i) \bullet \mu^{\delta_i}\right)^{x_j} \tag{12}$$

Send an audit request {H(id_i), σ_i} to the TPA.

(4) Send Challenge. TPA receives and uses the look up table to determine whether it is a valid identity. If it is an invalid user, the result is returned to user. If it is a legitimate user, the TPA sends the corresponding challenge to the cloud service provider.

The TPA randomly selects $v_i \in z_q^*$ and sends (13) to CSP.

$$\text{chal} = \left\{\left(H(id_i), v_i, \sigma_i\right)\right\}_{i\in I} \tag{13}$$

The cloud server uses the pseudorandom key π_k to compute

$$a_i = f_{\pi_k}(i, name) \tag{14}$$

thus restoring the original data m_i. According to a random value, calculating the m_i corresponding δ_i by the CSP.CSP aggregates

$$\sigma = \prod_{i\in I} \sigma_i^{v_i} \tag{15}$$

and calculates a linear combination of sampling blocks

$$u = \sum_{i\in I} v_i\delta_i \tag{16}$$

Then the CSP sends proof= (σ, u) to the TPA as an evidence of whether the data is complete.

(5) Verify. TPA receives proof= (σ, u) and verifies whether the equation is true. If the equation is satisfied, it means the data is complete, and then the TPA returns 1; else it returns 0.

(6) Members Join or Remove. When a member joins, the new user needs to register the corresponding account firstly and sends his identity to the RDC. RDC will redistribute the key

to the user. The user who gets the key will have the same rights as other users and he can perform data processing on the shared data. At the same time, RDC will also add this new user in the user identification table. When a user wants to leave the group, or if some malicious users are removed forcibly, RDC will mark the user's key as a special treatment. When a user with the same key logs in again, the user can no longer continue to view and modify the data.

(7) Members Modify Data and Achieve Identity Tracking. When the user wants to modify his own data, the user needs to send a request to the CSP. After the CSP authenticates, the CSP immediately informs the RDC and the RDC will use the dynamic broadcast list to broadcast in the group where the user is located. They can receive information about the data change. If there is no objection, the RDC will record the identity of the member. And the CSP will rereceive the user's modified data. When there is an argument about the operation of the data block m_i, the RDC can find the dishonest member by looking up the operation of the relevant user. The RDC finds the corresponding element by looking up the list (id_i, δ_i). Finally, it finds the cloud user M_j.

5. Security Analysis

In this section, we will prove the correctness, unforgeability, identity privacy protection, data confidentiality, and identity traceability of the scheme in detail. By certification we can make a conclusion that the proposed scheme has high security.

5.1. Correctness Analysis. In this paper, the correctness firstly means that a cloud user uploads data to a cloud server, after receiving permission from the RDC. We do this by applying for authentication. Only legitimate users can apply for this right. Malicious users are flagged and locked in time.

Secondly, correctness means that after a cloud user obtains reasonable authority and sends an audit request to the TPA, the TPA receives the evidence sent by the cloud service provider to perform data integrity audit. Therefore, the correctness of the scheme is that TPA can complete the integrity verification through the evidence provided by the cloud service provider, thus giving the cloud user an accurate answer to the data integrity audit. If the data is complete, the result is 1 and if the data is incomplete, 0 is returned. Now it is proved in detail as follows.

We can prove that the validation results are correct; that is, the left side of the equation equals the right.

$$\widehat{e}\left(\prod_{i\in I} H\left(id_i\right)^{v_i} \bullet u^{\mu}, PK\right) = \widehat{e}\left(g, \sigma\right) \tag{17}$$

Firstly, we simplify the equation

$$\begin{aligned}\sigma_i &= \sigma_i' \bullet \left(\frac{PK}{g^{x_j}}\right)^{-r_j} \left(H\left(id_i\right) \bullet \mu^{\delta_i}\right)^{x_j} \\ &= \left(H\left(id_i\right) \bullet \mu^{\delta_i} \bullet g^{r_j}\right)^{x_j'} \\ &= \left(H\left(id_i\right) \bullet \mu^{\delta_i} \bullet g^{r_j}\right)^{x_j'} \bullet \left(\frac{g^x}{g^{x_j}}\right)^{-r_j} \\ &\quad \bullet \left(H\left(id_i\right) \bullet \mu^{\delta_i}\right)^{x_j} = \left(\mathrm{H}\left(id_i\right) \bullet \mu^{\delta_i}\right)^{x}\end{aligned} \tag{18}$$

Secondly, we calculate

$$\begin{aligned}\widehat{e}\left(g, \sigma\right) &= \widehat{e}\left(g, \prod_{i\in I} \sigma_i^{v_i}\right) \\ &= \widehat{e}\left(g, \prod_{i\in I} \left(\left(\mathrm{H}\left(id_i\right) \bullet \mu^{\delta_i}\right)^{x}\right)^{v_i}\right) \\ &= \widehat{e}\left(g, \prod_{i\in I} \sigma_i^{v_i}\right) \\ &= \widehat{e}\left(g, \prod_{i\in I} \left(\left(\mathrm{H}\left(id_i\right) \bullet \mu^{\delta_i}\right)^{x}\right)^{v_i}\right) \\ &= \widehat{e}\left(g^x, \prod_{i\in I} \left(H\left(id_i\right)^{v_i} \bullet \mu^{\sum_{i\epsilon I} \delta_i v_i}\right)\right) \\ &= \widehat{e}\left(\prod_{i\in I} \left(H\ \left(id_i\right)^{v_i} \bullet \mu^{u}, PK\right)\right)\end{aligned} \tag{19}$$

The proof is over, so we can know that when the cloud server can save the data correctly, we can verify the integrity of the data through the evidence sent by the cloud service provider.

5.2. Unforgeability Analysis. Based on the security definition based on the discrete logarithm problem, we assume that there are malicious attackers who can falsify evidence and successfully authenticate with a third party. There must be an algorithm that solves the difficult problem of discrete logarithms based on the probability of nonnegligible. In order to complete the statement that the evidence in the scheme is not falsified now, we make the following game.

Game. We assume that there is shared data M. When a third party audit sends a challenge to the cloud service provider, challenge is $\{id_i, v_i\}$. The evidence generated by the original data is (σ, u) when the cloud based on the data M' $(\mathrm{M} \neq M')$; the service provider assumes that the evidence it generates is (σ, u'), and we specify $\mathrm{u} \neq u'$. If the TPA passes the integrity verification, then we say that cloud service providers have won this game.

When the cloud service provider wins the game, we can get the two TPA equations for verifying the data's integrity:

$$\widehat{e}\left(g, \sigma\right) = \widehat{e}\left(\prod_{i\in I} \left(H\ \left(id_i\right)^{v_i} \bullet \mu^{u}, PK\right)\right) \tag{20}$$

$$\widehat{e}\left(g, \sigma\right) = \widehat{e}\left(\prod_{i\in I} \left(H\ \left(id_i\right)^{v_i} \bullet \mu^{u'}, PK\right)\right) \tag{21}$$

Through the above two formulas, we know that g, μ are generators of G_1. And we know that PK=g^x, so PK is also a generator of group G_1. By applying the relevant properties of the bilinear map, we can infer the following equations:

$$\mu^u = \mu^{u'} \tag{22}$$

$$\mu = g^{r_1} PK^{r_2} \quad \left(r_1, r_2 \text{ from } Z_q.\right) \tag{23}$$

$$\mu^{\Delta u} = \left(g^{r_1} PK^{r_2}\right)^{\Delta u} = 1 \tag{24}$$

From the above three equations, we can infer that

$$PK = g^x = g^{-r_1 \Delta u / r_2 \Delta u} \tag{25}$$

From this, we can conclude that

$$x = \frac{-r_1 \Delta u}{r_2 \Delta u} \tag{26}$$

By observing the above formula, we find that the value of x can be solved when this equation is established, which is known from our previous game definition that $\Delta u \neq 0$. Therefore, the equation is meaningless only when r_2 is zero. We can calculate it. The probability of finding x in the group Z_q is 1-1/q. Since q is a large prime number, the probability of 1-1/q cannot be ignored. That is, when the cloud service provider wins this *g*ame, we can solve the problem of discrete logarithm with a nonnegligible advantage. This is contrary to the difficulty of discrete logarithm. Therefore, cloud service providers mentioned in the scheme can only pass the verification of the TPA if they provide the correct evidence, which illustrates that the proposed scheme has unforgeability.

5.3. Identity Privacy. As described in this scheme, the user's identity privacy means that when the TPA receives the audit request sent by the user, it cannot obtain the identity of the user from the audit request.

When we perform data integrity verification, we should pay attention to the protection of user's identity privacy. In the process of integrity auditing by a third party audit, the identity verification process hides the identity of the user by exploiting the good nature of the hash function so as to better protect the user's identity privacy. Specifically, on the one hand, during the integrity audit process, when the TPA authenticates the user, it is not necessary to directly compare the user's specific id value but rather compares the hash value stored by the third party audit center with itself. If the hash value shows that the user identity exists, then the identity of the sender of the audit request can be verified, and the third party audit center can send evidence to the cloud service provider. On the other hand, TPA cannot infer relevant information about the user's identity based on the audit request sent by the user.

5.4. Data Privacy. In the scheme proposed of this paper, the privacy of data refers to when a user sends a data authentication request: on the one hand, the information about user's data cannot be acquired by other parties except for the server; on the other hand, the user combines data. When the audit request is sent, the user's data information is not leaked out to the third party audit center during the processing of the audit request.

Table 4: Comparison of scheme features.

Features	Scheme [14]	Scheme [15]	Scheme [16]	Our scheme
data block identity privacy	√	√	√	√
Dynamic group	×	×	√	√
Identity tracking	×	×	×	√
Cloud user identity privacy	×	√	×	√

6. Results and Discussion

6.1. Algorithm Function Analysis. In cloud computing, data is usually shared by several users. Through comparative analysis of different schemes, as shown in Table 4, we can compare and analyze the different functions involved in the scheme, including identity tracking, data block privacy, dynamic groups, and identity privacy. Therefore, on the one hand, we can have a basic understanding of our scheme's function. On the other hand, we can better conduct the next step of research by comparing different schemes.

6.2. Algorithm Performance Analysis. In this section, we performed the following experiment. Based on these functions, we designed several experiments to assess the workload of involved entities. These experiments are carried out on a server running Linux OS with an Intel Pentium processor of 2.70GHZ and 4GB memory.

In terms of audit generation time efficiency, we evaluated the authentication algorithm. In terms of running time, we compared the efficiency of the three schemes (Yang [31], Ateniese G [32], and Wang [14]). The experimental results are shown in Figures 7 and 8. Our signature scheme is based on the BLS signature scheme and it is similar to the Yang [31] scheme. The scheme of Ateniese G [32] is based on proxy resignature. The computational cost is mainly the resignature of the data block and the modular exponent calculation on the G_1 group. The scheme of Wang [14] is based on RSA signatures. Its computational complexity is similar to that of ring signatures, and the amount of computation is also huge. It can be seen from the figure that Ateniese G [32] and Wang [14] are very time consuming, so our scheme has advantages.

We compare the time-consuming calculation with the number of other challenge blocks. The running time is shown in Figure 9. We can see the calculations of the three schemes, our scheme, Dongare D [33], and Yuan J and Yu [34]. The amount of computation for the three schemes is linear

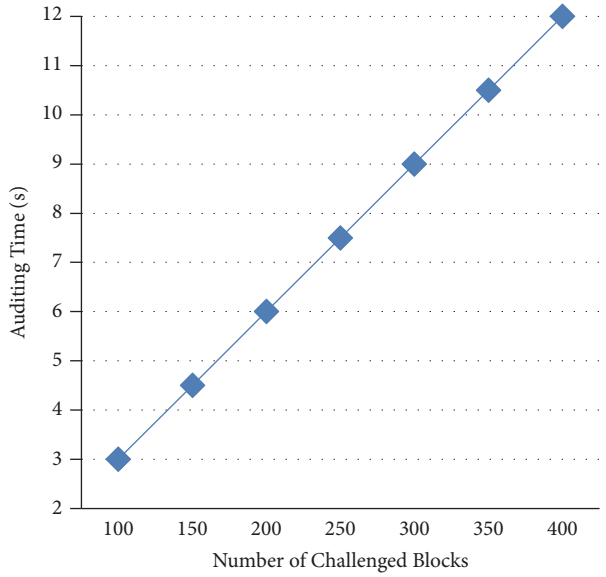

Figure 7: Audit request time.

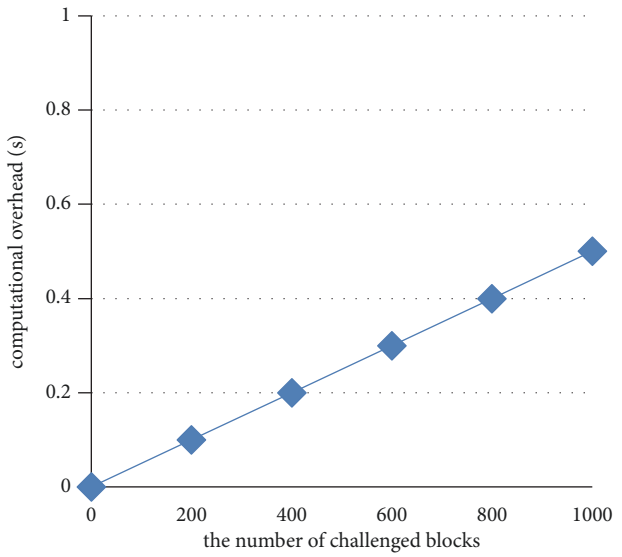

Figure 8: Challenge audit time.

with the number of data blocks being challenged increasing or decreasing. The more data blocks are challenged, the more time it takes to calculate. In the same experimental environment, our scheme spends less time than Yuan. J's scheme in calculating time. It takes more time than the Dongare D's scheme. However, this scheme can only achieve identity privacy; it cannot implement identity traceability. In terms of feasibility, our scheme has more obvious advantages. Specifically, generating a challenge message that specifies 400 random blocks takes only about 20 milliseconds, while the time specified as 1000 blocks increases to 50 milliseconds. The scheme meets the current mainstream cloud server configuration and it has strong feasibility.

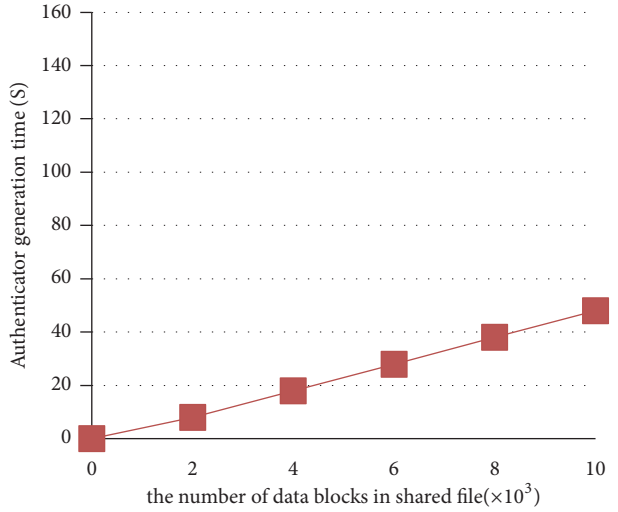

Figure 9: Authenticator generation time.

7. Conclusions

According to the above analysis, we can see that our proposed scheme is able to realize the desired security goals. In this paper, we establish a data sharing framework in cloud environment and propose a public auditing scheme with identity privacy and identity traceability for group members. The proposed auditing scheme achieves the security requirements that a well-constructed auditing scheme for shared cloud data should satisfy. As far as future work is concerned, we will continue to study how to improve the allocation of rights in the data integrity audit process and how to improve the security level of user data and protect identity privacy. The above will be the focus of our next research.

Data Availability

The data source of this paper is true and reliable. The relevant code link in this paper is https://github.com/xiaofeixue123/Integrity-audit.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61602287 and 61672330), Primary Research & Development Plan of Shandong Province (no. 2018GGX101037), and Major Scientific and Technological Innovation Project of Shandong Province (no. 2018CXGC0702).

References

[1] J. Lee, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2013.

[2] C. Erway, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 213–222, ACM, Chicago, Ill, USA, November 2009.

[3] B. Wang, B. Li, and H. Li, "Knox: privacy-preserving auditing for shared data with large groups in the cloud," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, pp. 507–525, Springer-Verlag, 2012.

[4] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[5] Q.-A. Wang, C. Wang, K. Ren, W.-J. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[6] C. Wang, Q. Wang, K. Ren et al., "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the IEEE INFO-COM*, vol. 62, pp. 525–533, IEEE, San Diego, Calif, USA, March 2010.

[7] L. Chen, "Using algebraic signatures to check data possession in cloud storage," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1709–1715, 2013.

[8] Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared big data," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[9] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1931–1940, 2017.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings of the IEEE INFOCOM*, pp. 534–542, IEEE, Piscataway, NJ, USA, March 2010.

[11] B. Liang, J. Y. Cao, Y. Z. Qin et al., "Survey of proofs on data storage security in cloud computing," *Application Research of Computers*, vol. 29, no. 7, pp. 2416–2421, 2012.

[12] G. Z. Qin, K. S. Wu, and H. Xiong, "A review on data integrity auditing protocols foe data storage in cloud computing," *Net Info Security*, pp. 1–6, 2014.

[13] S. H. Wang, D. W. Chen, Z. W. Wang et al., "A new solution of privacy preserving public auditing scheme foe cloud storage security," *Telecommunications Science*, vol. 28, no. 9, pp. 15–21, 2012.

[14] B. Wang, B. Li, and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," in *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD '12)*, pp. 295–302, IEEE Computer Society, June 2012.

[15] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Lightweight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," *Journal of Network and Computer Applications*, vol. 82, pp. 56–64, 2017.

[16] B. Wang, H. Li, and M. Li, "Privacy-preserving public auditing for shared cloud data supporting group dynamics," in *Proceedings of the ICC 2013 - 2013 IEEE International Conference on Communications*, pp. 1946–1950, Budapest, Hungary, June 2013.

[17] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in *Proceedings of International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pp. 319–333, Springer-Verlag, London, UK, 2009.

[18] W. Luo and G. Bai, "Ensuring the data integrity in cloud data storage," in *Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS2011*, pp. 240–243, IEEE, China, September 2011.

[19] R. Curtmola, O. Khan, and R. Burns, "Robust remote data checking," in *Proceedings of the 4th ACM International Workshop*, pp. 63–68, ACM, Alexandria, Va, USA, October 2008.

[20] Y. Zhang and M. Blanton, "Efficient dynamic provable possession of remote data via balanced update trees," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013*, pp. 183–194, China, May 2013.

[21] H. Wang and Y. Zhang, "On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 264–267, 2014.

[22] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231–2244, 2012.

[23] Y. Zhu, H. Wang, Z. Hu, G. Ahn, H. Hu, and S. S. Yau, "Efficient provable data possession for hybrid clouds," in *Proceedings of the 17th ACM Conference*, pp. 756–758, ACM, Chicago, Ill, USA, October 2010.

[24] Y. Zhu, H. Hu, G. Ahn, Y. Han, and S. Chen, "Collaborative integrity verification in hybrid clouds," in *Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 191–200, Miami, Fla, USA, October 2011.

[25] F. Zafar, A. Khan, U. R. S. Malik et al., "A survey of cloud computing data integrity schemes: design challenges, taxonomy and future trends," *Computers & Security*, 2016.

[26] K. Han, Q. Li, and Z. Deng, "Security and efficiency data sharing scheme for cloud storage," *Chaos Solitons & Fractals the Interdisciplinary Journal of Nonlinear Science & Nonequilibrium & Complex Phenomena*, vol. 86, pp. 107–116, 2016.

[27] T. Jiang, X. Chen, and J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2363–2373, 2016.

[28] L. Xue, J. Ni, Y. Li, and J. Shen, "Provable data transfer from provable data possession and deletion in cloud storage," *Computer Standards & Interfaces*, vol. 54, pp. 46–54, 2017.

[29] D. Sánchez and M. Batet, "Privacy-preserving data outsourcing in the cloud via semantic data splitting," *Computer Communications*, vol. 110, pp. 187–201, 2017.

[30] Y. Yu, J. Ni, W. Wu et al., "Provable data possession supporting secure data transfer for cloud storage," in *Proceedings of the 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, pp. 38–42, IEEE, Krakow, Poland, 2016.

[31] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *The Journal of Systems and Software*, vol. 113, pp. 130–139, 2016.

[32] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 598–609, ACM, Virginia, Va, USA, November 2007.

[33] D. Dongare and V. Kadroli, "Panda: Public auditing for shared data with efficient user revocation in the cloud," in *Proceedings of the 2016 Online International Conference on Green*

Engineering and Technologies (IC-GET), pp. 2904–2912, IEEE, Coimbatore, India, November 2016.

[34] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-user modification," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 2121–2129, Canada, May 2014.