Hindawi
Security and Communication Networks
Volume 2022, Article ID 7302767, 9 pages
https://doi.org/10.1155/2022/7302767

Research Article

Improved Efficient Privacy-Preserving Certificateless Provable Data Possession Scheme for Cloud Storage

Xiuguang Li,[1,2] **Ruifeng Li**,[2] **Xu An Wang**,[2] **Ke Niu**,[2] **Hui Li**,[1] **and Xiaoyuan Yang**[2]

[1]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China*
[2]*Chinese People's Armed Police Force Engineering University, Xi'an, China*

Correspondence should be addressed to Hui Li; lihui@mail.xidian.edu.cn

Received 2 June 2022; Accepted 15 July 2022; Published 30 August 2022

Academic Editor: Yinbin Miao

Copyright © 2022 Xiuguang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cloud storage technology is evolving at a high speed; effectively auditing the cloud data's integrity has become a focal point. Recently, Ming and Shi proposed a certificateless integrity auditing scheme with a privacy protection function. The scheme used the certificateless cryptosystem to solve the certificate management problem of the auditing schemes based on public key infrastructure and the key escrow problem of the identity-based auditing schemes. Although their scheme is novel and efficient, we found that their scheme was not secure and could not achieve integrity auditing of cloud data. The malicious cloud server can generate the proof through the blocks and tags sent by the user. On the basis of the original scheme, we propose an improved auditing scheme; our new scheme is more secure and effective. In addition, for the problem of idle tags in the existing cloud data integrity auditing scheme, we propose the idea of intermediate tags and we applied the idea to the improved scheme to improve audit efficiency.

1. Introduction

Users are increasingly inclined to store data in the cloud to obtain more convenient data management services. Cloud service providers (CSP) centrally hold massive amounts of users' data. For an attacker, a successful attack on the cloud server will gain a great deal. Therefore, it is easy for CSPs to become the targets of centralized attacks. The dishonest CSPs may also deliberately delete users' data to reduce their own storage burden or deliberately conceal security incidents that damage data integrity to maintain their own reputation. Therefore, the cloud data integrity audit schemes are proposed to effectively solve problems [1].

Motivation: We note that the existing audit schemes require users to calculate data tags corresponding to all data blocks when preprocessing data blocks and upload them to the CSP for storage. However, in the auditing process, generally few tags are used to generate the proof. Once the proof is verified, it can ensure that each data block specified by the auditor is complete and guarantee all original data's integrity with a high confidence probability. For 1,000,000 data blocks with a size of 4 kB, assuming that the server deletes or is tampered with 1% of the data blocks, the auditor only needs to audit 460 data blocks, which can be higher than 99% confidence probability [2] to judge the integrity of all the data. Therefore, most of the data tags are idle during the audit process. Suppose it is an application scenario where data blocks are frequently updated [3]; a large number of tags are calculated and stored in the cloud, but they will be updated as the data blocks are updated before they are used, resulting in larger computing and storage resources waste. To solve the idle tags problem, we propose the idea of intermediate tags. Before uploading the data, when processing the data, users only generate the key intermediate tags and then upload them to CSP. When the third-party auditor (TPA) challenges the cloud data, CSP generates complete certification tags for the challenged data block. Then they enter the normal audit process.

Recently, Ming and Shi [4] proposed a certificateless auditing scheme called CLPDP that supports privacy protection. In their scheme, CSP can use tags to easily forge the proof. Even if all outsourced data are deleted by CSP, it can

still give the correct proof to pass the audit. So we point out the security problem in their scheme. In addition, we find that the original scheme is one that can apply the idea of intermediate tags. Therefore, we also improved the original scheme.

Our contributions are as follows:

(1) We analyze Ming and Shi's scheme and find the security problem. CSP can forge the proof to pass the audit. Then we described the attack method in detail.

(2) We propose the idea of intermediate tags, which can reduce the computing overhead of users in audit schemes. After improving the safety of the original scheme, we use the idea of intermediate tags to promote the original scheme.

(3) We performed security analysis on the improved scheme, and we proved that the improved scheme is secure. The efficiency of the improved scheme is also analyzed and compared with that of the original scheme. The improved scheme is more efficient, which proves the applicability of intermediate tags.

2. Related Works

Early data integrity audit schemes required users to download all their stored data and verify the downloaded data locally. However, most users store a large amount of data, so it requires high communication, storage, and computing costs for users to download all data for verification, and users generally cannot meet such requirements. Ateniese et al. [2] formally defined the Provable Data Possession (PDP) scheme. When verifying the integrity, users divide data files into blocks, and only partial data blocks are downloaded. Finally, the integrity of all data can be verified with a very high confidence rate. This method enables users to complete the audit task without downloading complete files, reducing the huge communication cost in the process of a data integrity audit. In 2013, Wang et al. [5] introduced TPA into the data integrity audit system. Users can further reduce their own expenses by outsourcing audit tasks to TPA. At present, scholars add various functions to the basic data integrity audit scheme [2] to meet the requirements of different application scenarios.

Users will inevitably need to change their data after uploading data files. Therefore, the cloud data's content should be allowed to change dynamically. Considering the urgent need for data integrity audit schemes in the dynamic update, scholars put forward audit schemes with dynamic update functions. Dynamic data update has gradually become the basic function in cloud data integrity audit schemes, which is indispensable in the application of real scenarios. Existing data structures applied to dynamic data updates mainly include Index Switcher, Index Hash Table, Merkle Hash Tree, Skip List, Dynamic Hash Table, Red-Black Tree, etc. In the construction of a data integrity verification scheme supporting dynamic data updates, the difficulty lies in solving the problem of extra computation costs caused by index change.

Jin et al. [6] constructed the mapping from the data block index to the tag index and designed the Index Switcher data structure to avoid the extra computational overhead caused by tag recalculation. In addition, the dispute arbitration function is added to the proposed audit scheme to ensure that users or the cloud will not commit improper acts during the audit process. Tian et al. [7] proposed the audit scheme supporting dynamic updates, privacy protection, and batch audit. The dynamic hash table data structure is designed to realize fast audit and efficient data updates by recording the attributes of files and data blocks at the audit ends. Shen et al. [8] proposed the whole/sampling audit method to solve the problem of distrust between users and the cloud and designed a double-linked information table to achieve efficient data update. Their scheme also supported the batch audit function. Guo et al. [9] constructed a multileaf authentication method based on the Merkle tree, which can simultaneously authenticate multiple leaf nodes and corresponding indexes and realize batch data updates. The scheme supports log auditing. By checking the log files generated by auditors, users can verify whether the auditors perform their audit work honestly. The public audit protocol designed by Hou et al. [10] supports blockless verification and batch verification. The scheme uses the chameleon authentication tree to realize the efficient and dynamic operation of outsourced data and reduces computing costs and improves the audit efficiency. Mishra et al. [11] used a binomial binary tree and indexed hash table data structure to construct an audit scheme supporting batch audit and efficient dynamic update based on BLS signature.

The reliability of data is the basis of its value and benefit. After the reliability of data is solved, other problems of data such as consistency, practicality, and availability are meaningful. Multicopy storage is the most straightforward and simple way to improve reliability. CSP provides storage services at low prices. Users can use the massive storage space it provides. More and more users choose multicopy storage to obtain more availability of data. The audit schemes supporting dynamic manipulation of multiple replicas while ensuring data integrity remain to be explored and further investigated. Curtmola et al. [12] constructed the first multicopy audit scheme, in which each copy can generate a corresponding integrity proof against challenges, and storing multiple copies is more efficient than storing each copy individually. Liu et al. [13] constructed the multiple-copy audit scheme supporting data dynamic updating. The Merkle hash tree node used in their scheme contains the node level parameters, which are allocated to each data block. It is more efficient when verifying multiple replica updates. The audit scheme of Guo et al. [14] reduces the storage burden of CSP by sharing an authenticated identity tree among multiple copies. The scheme supports multicopy and batch auditing, which also reduces the computational cost. Yaling and Li [15] proposed a flexible multicopy PDP scheme based on the characteristics of a multibranch tree. Their scheme ensures the integrity and reliability of multiple copies and implements the verification of any copy and supports dynamic update operation and privacy protection.

In recent years, in order to optimize audit performance and improve update efficiency, batch audit and batch update have become indispensable functions of cloud data integrity audit schemes. Qi et al. [16] applied the rank-based Merkle hash balanced tree to integrity verification and improved the dynamic update's efficiency. Deng et al. [17] implemented batch auditing using BLS signature and rank-based Merkle hash tree.

Later, scholars introduce TPA to perform a public audit on behalf of users to reduce the computation cost. However, TPAs are often not fully trusted [18], which can lead to the disclosure of users' privacy [19]. Li et al. [20] solved the key management problem based on fuzzy identity. The scheme took the user's biometrics as the identity and designed a corresponding audit protocol to protect the data content. Wang et al. [21] scheme uses a ring signature to calculate the metadata required for verification. The authenticator and random mask technology are used to protect data privacy; the scheme can also realize batch audits. The audit scheme of Wang et al. [22] is based on an algebraic signature and integrates forward error correction codes to enhance data possession assurance and recover data when a small number of blocks are deleted, thus significantly reducing communication complexity.

With the development of blockchain technology, many scholars apply blockchain technology to cloud data integrity audits [23]. The certificateless audit scheme proposed by Zhang et al. [24] can resist malicious TPA; the scheme uses Bitcoin as the source of pseudorandom numbers to help generate challenging information. Li et al. [25] proposed a lightweight audit scheme with blockchain technology for integrity audit. In their scheme, the user and CSP are set as two mutually untrustworthy entities, and the TPA is removed. After the user stores the lightweight verification tags into the blockchain, the Merkle hash tree is constructed through the tags to generate the proof, so as to save computational power. Yang et al. [26] provided the mutual blockchain for outsourced cloud data and proposed an incentive mechanism based on credit, so that CSPs can supervise each other, which prevents collusion and realizes public audit efficiently. Yang et al. [27] proposed a certificateless multicopy and multicloud data public audit scheme based on blockchain technology. Their scheme leverages the unpredictability of blocks in the blockchain to build fair challenge information, preventing malicious auditors from colluding with CSP to deceive users. Wang et al. [28] used blockchain to replace TPAs and designed a blockchain-based fair payment smart contract for a cloud data audit. In their scheme, users and CSP will run blockchain-based smart contracts to ensure that the cloud periodically submits data to the cloud with proof of possession. Only after verification can the CSP be paid. Wei et al. [29] built a blockchain integrity protection mechanism. The scheme deploys the distributed virtual machine agent model on the cloud allowing multitenant collaboration and achieving reliable storage, monitoring, and verification tasks. Reference [30] proposed a protection model based on a private chain, which synchronously uploads modification records of files and hash values of files to blockchain for storage and judges whether the data is complete by comparing hash values.

Quantum computers use qubits to represent many possible states of 1 and 0 at the same time and have more processing power than standard computers. Most cloud storage data auditing schemes are based on a traditional cryptosystem. However, with the introduction of algorithms such as quantum large number decomposition, the traditional cryptosystem loses its security. Lattice-based cryptography is generally considered to be effective against the quantum attack. Xu et al. [31] designed the first lattice-based cloud data audit scheme based on the small integer solution problem. The audit scheme designed by Liu and Cao [32] supports public verification but does not provide strict security certification. Zhang et al. [33] designed an ID-based public audit protocol based on lattice by using ID-based signature technology and further provided a solution to solve the key exposure problem [34], which protected user data's privacy. In addition, TPA cannot obtain information about users' data during audit verification. Sasikala and Shoba Bindu [35] designed a lattice-based certificateless public auditing protocol for the first time, but it was pointed out by [36] that the scheme had security problems.

Organization: We organize our paper as follows. In Section 3, we reviewed the certificateless privacy protection secure cloud storage scheme of Ming and Shi. Section 4 describes the attack against the original scheme. In Section 5, we propose the concept of intermediate tags and give an improved audit protocol. In Section 6, the security and performance of the improved scheme are analyzed to prove that it is safer and more efficient. Finally, in Section 7, we summarize our work.

3. Review of Ming and Shi's Scheme

The system model of Ming and Shi is shown in Figure 1, including a key generation center (KGC), a data owner (DO), CSP, a data user (DU), and TPA. Figure 1 shows their system model. To facilitate understanding, we define and explain the various symbols and variables that appear in our paper in Table 1.

Specifically, the following is the operation process of the original scheme:

(1) Setup: KGC first selects the cyclic group G on the elliptic curve E, defines the large prime number q with the order of G, and selects the generator $P \in G$. Then it selects the secure hash function $H_{1,2,3,4}: \{0,1\}^* \longrightarrow Z_q^*$, selects a random $\lambda \in Z_q^*$ as the system master key, and calculates $P_{\text{pub}} = \lambda \cdot P \in G$. Finally, KGC keeps the master key in secret and exposes the parameters.

(2) PartialKeyGen: DO sends the real identity information $\text{ID} \in Z_q^*$ to KGC. After KGC receives $\text{ID} \in Z_q^*$, it selects a random number $u \in Z_q^*$ and calculates $PID_1 = u \cdot P$, $PID_2 = I\ D \oplus H_1(u \cdot P_{\text{pub}} \| PID_1)$.

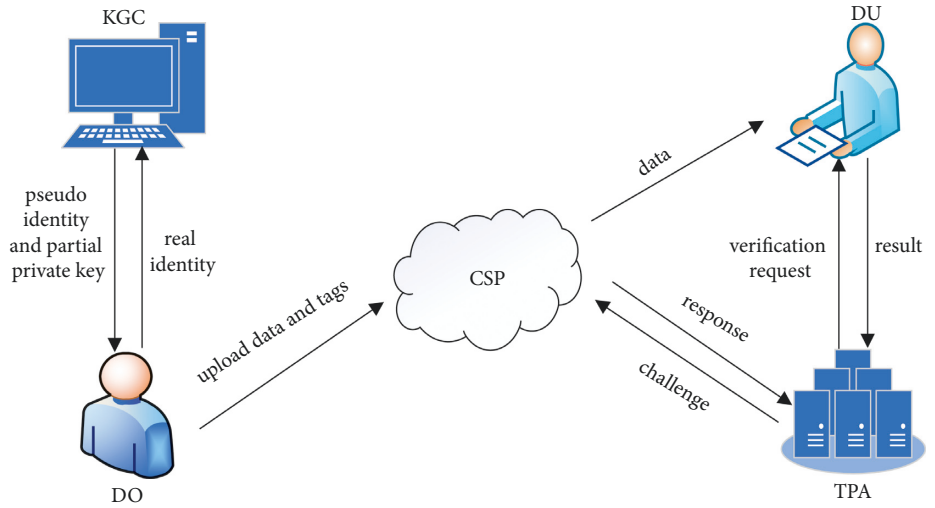

Figure 1: The system model.

Table 1: Notations.

Notations	Descriptions
E	The elliptic curve
F_p	The finite field
G	The cyclic group
P	Generator of G
$H_{1,2,3,4}$	Secure hash function $H(\cdot)$: $\{0,1\}^* \longrightarrow Z_q^*$
P_{pub}	Public key of the system
$\text{ID} \in Z_q^*$	DO's real identity
PID	DO's virtual identity
λ, u, d, r_i, v_i	Random numbers
(x, y)	DO's secret key
(D, X)	DO's public key
id_i	The identifier of the data block m_i
m_i	The data block
ω_j, ϕ_j, τ	Intermediate parameters
R_i, s_i	The tags of m_i
Q	The collection of challenged indexes
$\{\alpha, \beta\}$	The proof
H	The computational cost of one hash
A_Z	The computational cost of one addition on Z_q^*
M_Z	The computational cost of one multiplication on Z_q^*
A_G	The computational cost of one point addition on G
M_G	The computational cost of one point multiplication on G

Then KGC sends DO's virtual identity $PI\ D = \{PID_1, PID_2\}$ to DO and randomly selects $d \in Z_q^*$.

alculates $D = d \cdot P$, $\tau = H_2(PI\ D \| D)$, $y = d + \lambda \cdot \tau$. Finally, KGC sends DO's partial keys $\{D, y\}$ to DO.

(3) SecretValueGen: DO randomly chooses $x \in Z_q^*$ and obtains complete private key $\{x, y\}$.

(4) PublicKeyGen: DO calculates $X = x \cdot P$ and obtains the complete public key $\{D, X\}$.

(5) TagGen: the data file M is divided into n blocks by DO as $M = \{m_1, m_2, \ldots, m_n\}$, where $m_{i(1 \le i \le n)} \in Z_q^*$. DO selects a random number $r_i \in Z_q^*$ and calculates $R_i = r_i P$, $w_i = H_3(X \| R_i \| id_i)$, $\phi_i = H_4(D \| R_i \| id_i)$, and $s_i = r_i \cdot m_i + w_i \cdot x + \phi_i \cdot y$ for $i \in \{1, 2, \ldots, n\}$, where id_i is the identifier of m_i. Thus the tags $\sigma = \{R_1, R_2, \ldots, R_n, s_1, s_2, \ldots, s_n\}$ are generated by DO; they are sent with the data blocks to CSP. DO deletes the local data and tags.

(6) Challenge: after receiving DU's audit request, TPA generates the challenge message. It first selects a random subset Q in $\{1, 2, \ldots, n\}$. The subset Q includes c elements. For $j \in Q$, TPA randomly selects $v_j \in \mathbb{Z}_q^*$; then it sends $\text{chal} = \{j, v_j\}_{j \in Q}$ as the challenge message to CSP.

(7) ProofGen: CSP calculates $\alpha = \sum_{j \in Q} v_j \cdot s_j \cdot P$ and $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$ after it receives $\text{chal} = \{j, v_j\}_{j \in Q}$; then it sends $\{\alpha, \beta\}$ as the proof to TPA.

(8) Verify: after receiving $\{\alpha, \beta\}$, TPA calculates $\tau = H_2(PI\ D \| D)$. Then it calculates $\omega_j = H_3(X \| R_j \| id_j)$ and $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$ and verifies

$$\alpha \stackrel{?}{=} \beta + \left(\sum_{j \in Q} \omega_j \cdot v_j\right) \cdot X + \left(\sum_{j \in Q} \phi_j \cdot v_j\right) \cdot \left(D + \tau \cdot P_{\text{pub}}\right). \tag{1}$$

If equation (1) holds, DO's data is complete.

The proof of the correctness of equation (1) is as follows:

$$\begin{aligned}\alpha &= \sum_{j\in Q} v_j s_j P = \sum_{j\in Q} v_j\left(r_j m_j + \omega_j x + \phi_j y\right)P \\ &= \sum_{j\in Q} v_j r_j m_j P + \sum_{j\in Q} v_j \omega_j x P + \sum_{j\in Q} v_j \phi_j y P \\ &= \sum_{j\in Q} v_j m_j R_j + \left(\sum_{j\in Q} v_j \omega_j\right)X + \left(\sum_{j\in Q} v_j \phi_j\right)\left(D + \tau P_{\text{pub}}\right) \\ &= \beta + \left(\sum_{j\in Q} \omega_j v_j\right)X + \left(\sum_{j\in Q} \phi_j v_j\right)\left(D + \tau P_{\text{pub}}\right).\end{aligned} \tag{2}$$

4. Our Attack

In the scheme of Ming and Shi, we find that the CSP can calculate the value of the aggregated data blocks needed at the Proofgen stage. In this way, even if the CSP deletes DO's cloud data, the correct data possession proof can be generated by it at the Proofgen stage and passed the audit. In this section, we show two types of attacks; the process by which CSP forges the "correct" blocks is also introduced.

4.1. The First Type of Attack. The first attack is caused by a design error in the verification equation; the detailed description is as follows:

Assume that the entities in the scenario run the audit scheme following the process described above; when the scheme progresses to the ProofGen stage, CSP needs to generate the proof $\{\alpha, \beta\}$. We note that, in equation (1), CSP can obtain all values except α and β, so CSP just needs to randomly select $\alpha \in G$; it can obtain β by calculating equation (1). Similarly, CSP can also calculate the value of α by calculating equation (1) when it randomly selects $\beta \in G$. Thus, CSP does not need to store DO's data to generate the proof $\{\alpha, \beta\}$ that satisfies equation (1).

4.2. The Second Type of Attack. At the TagGen stage, the CSP receives blocks and tags. CSP first calculates $s_i' = s_i \cdot P$, so it gets the following equations:

$$\begin{cases} s_1' = r_1 m_1 \cdot P + w_1 \cdot X + \phi_1 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ s_2' = r_2 m_2 \cdot P + w_2 \cdot X + \phi_2 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ \vdots \\ s_n' = r_n m_n \cdot P + w_n \cdot X + \phi_n \cdot \left(D + \tau \cdot P_{\text{pub}}\right). \end{cases} \tag{3}$$

X and D are DO's public keys, CSP knows the values of X, D, and P_{pub}, it can also calculate the value of w_i and ϕ_i for $1 \le i \le n$, and then it obtains $r_i m_i$ for $1 \le i \le n$ to calculate the following equations:

$$\begin{cases} r_1 m_1 \cdot P = s_1' - w_1 \cdot X - \phi_1 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ r_2 m_2 \cdot P = s_2' - w_2 \cdot X - \phi_2 \cdot \left(D + \tau \cdot P_{\text{pub}}\right), \\ r_n m_n \cdot P = s_n' - w_n \cdot X - \phi_n \cdot \left(D + \tau \cdot P_{\text{pub}}\right). \end{cases} \tag{4}$$

At the ProofGen stage, the CSP needs to calculate

$$\beta = \sum_{j\in Q} v_j \cdot m_j \cdot R_j = \sum_{j\in Q} v_j \cdot m_j r_j P. \tag{5}$$

Even if CSP deletes $\{m_1, m_2, \ldots, m_n\}$, it can calculate the value of β with $r_1 m_1 \cdot P$, $r_2 m_2 \cdot P \ldots\ldots r_n m_n \cdot P$, which can pass the audit.

5. The Improved Auditing Scheme

In this section, we first explain what an intermediate tag is and how to set an intermediate tag; then we give an improved secure auditing scheme.

We first analyze the probability of misbehavior detection in existing PDP schemes. For $n = 1000000$ 4 KB data blocks, we assume that 1% of the data blocks' integrity is damaged; TPA can specify 460 data blocks to obtain a confidence probability higher than 99%. We set n as the data blocks' total number, c_1 as damaged data blocks' number, and c_2 as randomly challenged data blocks' number during the audit. We set a random variable X representing the number of corrupted blocks in the challenged blocks; P_X represents the corresponding probability. We have the deduction as follows:

$$\begin{aligned}P_X &= P\{X \ge 1\} = 1 - P\{X = 0\} = 1 \\ &\quad - \frac{n - c_1}{n} \cdot \frac{n - 1 - c_1}{n - 1} \cdots \frac{n - c_2 + 1 - c_1}{n - c_2 + 1}.\end{aligned} \tag{6}$$

Because $(n - c_1/n) > (n - 1 - c_1/n - 1)$, so:

$$P_X \ge 1 - \left(\frac{n - c_1}{n}\right)^{c_2}. \tag{7}$$

In the case of $c_1/n = 1\%$, when c_2 is 300, 460, and 688, P_X is greater than 95%, 99%, and 99.9%, respectively. Therefore, in an audit process, few data blocks are challenged, and the relevant tags are used to generate the proof. Most of the other data blocks and relevant tags are idle.

Assuming that there are total $n = 1000000$ data blocks and tags stored in the cloud, 460 of them are challenged in each audit, and the challenged data blocks are different in multiple audits. Then it takes about 2173 audit times to use all the data blocks and corresponding tags. In practical applications, due to the user's demand for data update, many idle blocks and corresponding tags are modified and updated before they can be used, resulting in a large waste of computing overhead.

Therefore, we propose the idea of intermediate tags: at the TagGen stage, users only generate intermediate tags composed of the private key and data blocks, instead of calculating mature tags used by CSP when generating evidence, which reduces the calculation overhead of users. At the ProofGen stage, CSP calculates mature tags of only a few

challenged data blocks according to the challenge information from the TPA and uses them to generate the proof. The idea of intermediate tags is applied to the following improved scheme:

(1) Setup: KGC first selects the cyclic group G on the elliptic curve E, defines the large prime number q with the order of G, selects $P \in G$ as the generator, $H_{1,2,3,4}: \{0,1\}^* \longrightarrow Z_q^*$ as hash functions, and a random $\lambda \in Z_q^*$ as the system master key, and calculates $P_{\text{pub}} = \lambda \cdot P \in G$. Finally, KGC keeps λ in secret and exposes the public parameters.

(2) PartialKeyGen: DO sends the real identity information $\text{ID} \in Z_q^*$ to KGC. After KGC receives $\text{ID} \in Z_q^*$, it selects a random number $u \in Z_q^*$ and calculates $\text{PID}_1 = u \cdot P$, $PID_2 = I\ D \oplus H_1(u \cdot P_{pub} \| PID_1)$. Then KGC sends DO's virtual identity $PI\ D = \{PID_1, PID_2\}$ to DO and randomly selects $d \in Z_q^*$ and calculates $D = d \cdot P$, $\tau = H_2(PI\ D \| D)$, $y = d + \lambda \cdot \tau$. Finally, KGC sends DO's partial keys $\{D, y\}$ to DO.

(3) SecretValueGen: DO randomly chooses $x \in Z_q^*$ and obtains complete private key $\{x, y\}$.

(4) PublicKeyGen: DO calculates $X = x \cdot P$ and obtains the complete public key $\{D, X\}$.

(5) TagGen: the data file M is divided into n blocks by DO as $M = \{m_1, m_2, \ldots, m_n\}$, where $m_{i(1 \le i \le n)} \in Z_q^*$. DO randomly selects $r_i \in Z_q^*$, $k \in Z_q^*$ and calculates $R_i = r_i P$, $s_i = r_i m_i + k$. Note that here we have simplified the formula for calculating s_i, and the intermediate tag s_i in the improved scheme is different from the mature tag s_i in the original scheme. Thus the tags $\sigma = \{R_1, R_2, \ldots, R_n, s_1, s_2, \ldots, s_n\}$ are generated by DO; they are sent with the data blocks to CSP. DO sends k to TPA and deletes the local data and tags.

(6) Challenge: after receiving DU's audit request, TPA generates the challenge message. It first selects a random subset Q in $\{1, 2, \ldots, n\}$. The subset Q includes c elements. For $j \in Q$, TPA randomly chooses $v_j \in \mathbb{Z}_q^*$; then it sends $\text{chal} = \{j, v_j\}_{j \in Q}$ to CSP as the challenge message.

(7) ProofGen: CSP calculates $w_j = H_3(X \| R_j \| id_j)$ and $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$ after receiving $chal = \{j, v_j\}_{j \in Q}$. Then it calculates $\alpha = \sum_{j \in Q} v_j \cdot (s_j \cdot P + w_j X + \phi_j Y)$, $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$ as the proof and sends $\{\alpha, \beta\}$ to TPA.

(8) Verify: after receiving the proof $\{\alpha, \beta\}$, TPA calculates $\tau = H_2(PI\ D \| D)$ and calculates $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$. Then, it verifies

$$\alpha \stackrel{?}{=} \beta + \sum_{j \in Q} v_j kP + \sum_{j \in Q} w_j v_j X + \sum_{j \in Q} \phi_j v_j \left(D + \tau P_{\text{pub}}\right). \tag{8}$$

If equation (8) holds, DO's cloud data is complete. The correctness of equation (8) is as follows:

$$\begin{aligned}
\alpha &= \sum_{j \in Q} v_j \cdot \left(s_j \cdot P + w_j X + \phi_j \left(D + \tau P_{\text{pub}}\right)\right) + \sum_{j \in Q} v_j kP \\
&= \sum_{j \in Q} v_j \left(r_j \cdot m_j + w_j x + \phi_j \cdot y\right) \cdot P + \sum_{j \in Q} v_j kP \\
&= \sum_{j \in Q} v_j r_j m_j P + \sum_{j \in Q} v_j w_j x P + \sum_{j \in Q} v_j \phi_j y P + \sum_{j \in Q} v_j kP \\
&= \sum_{j \in Q} v_j m_j R_j + \left(\sum_{j \in Q} v_j w_j\right) X + \left(\sum_{j \in Q} v_j \phi_j\right) \left(D + \tau P_{\text{pub}}\right) + \sum_{j \in Q} v_j kP \\
&= \beta + \sum_{j \in Q} v_j kP + \left(\sum_{j \in Q} w_j v_j\right) X + \left(\sum_{j \in Q} \phi_j v_j\right) \left(D + \tau P_{\text{pub}}\right).
\end{aligned} \tag{9}$$

6. Analysis of the Improved Protocol

In this section, we first demonstrate that the improved scheme can resist the above attacks. Then the improved scheme's performance is analyzed. We also compare the computation overhead in two schemes, so as to prove that our improved scheme is more efficient.

6.1. Security Analysis. CSP holds the following equations in the improved scheme:

$$\begin{cases}
s_1 = r_1 m_1 + k, \\
s_2 = r_2 m_2 + k, \\
\vdots \\
s_i = r_i m_i + k.
\end{cases} \tag{10}$$

In equation (10), r_i and k are unknown to CSP; it always has more unknowns than equations, so CSP cannot solve the equations to calculate the values of r_i and k. At the ProofGen stage, CSP cannot know $r_1 m_1, r_2 m_2, \ldots, r_n m_n$. When CSP uses the second of the above attacks, it can list the following equations:

Table 2: The computational costs of the two schemes at each stage.

	The original scheme	The improved scheme
TagGen	$nM_G + 2nH + 3nM_Z + 2nA_Z$	$nM_G + nM_Z + nA_Z$
ProofGen	$(c-1)A_G + 2cM_Z + (c-1)A_Z + cM_G$	$2cH + (c-1)A_G + (5c+1)M_Z + (4c-4)A_Z + (c+4)M_G$
Verify	$(2c+1)H + 3A_G + 2cM_Z + (2c-2)A_Z + 3M_G$	$(2c+1)H + 4A_G + (3c+1)M_Z + (3c-3)A_Z + 4M_G$

$$\begin{cases} m_1 \cdot R_1 = s_1 \cdot P + k \cdot P, \\ m_2 \cdot R_2 = s_2 \cdot P + k \cdot P, \\ \vdots \\ m_i \cdot R_i = s_i \cdot P + k \cdot P. \end{cases} \tag{11}$$

Since CSP does not know the value of k, it cannot compute the value of $m_i \cdot R_i$. When generating $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$, CSP can not calculate the value of β with the tag uploaded by DO. Only when the $m_1, m_2, \ldots, m_n$ are stored correctly and completely by CSP, can CSP generate the correct β and pass the TPA audit.

When CSP uses the first of the above attacks, after randomly selecting one of the values of α and β, it attempts to obtain the value of the other variable by calculating equation (1). But in equation (9), k is unknown to CSP, and CSP cannot compute β from α and equation (1) or compute α from β and equation (1).

6.2. Performance Analysis. The idea of intermediate tags is to save computing overhead for DO. The difference of storage and communication costs between two schemes is small, so we mainly analyze the computing costs of the two schemes.

In the original scheme, at the TagGen stage, DO needs to calculate $R_i = r_i P$, $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$, $s_i = r_i \cdot m_i + w_i \cdot x + \phi_i \cdot y$, and the calculation cost is $nM_G + 2nH + 3nM_Z + 2nA_Z$. At the ProofGen stage, CSP calculates $\alpha = \sum_{j \in Q} v_j \cdot s_j \cdot P$ and $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$, set c as the number of elements in Q, and the calculation cost is $(c-1)A_G + 2cM_Z + (c-1)A_Z + cM_G$. At the Verify stage, TPA calculates $\tau = H_2(PI\ D \| D)$, for $j \in Q$, calculate $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ and equation (1), the and calculation cost is $(2c+1)H + 3A_G + 2cM_Z + (2c-2)A_Z + 3M_G$.

In the improved scheme, at the TagGen stage, DO only needs to calculate $R_i = r_i P$, $s_i = r_i m_i + k$, and the computational cost is $nM_G + nM_Z + nA_Z$. At the ProofGen stage, CSP calculates $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ for each challenged block. Then, it calculates $\alpha = \sum_{j \in Q} v_j \cdot (s_j \cdot P + w_j X + \phi_j (D + \tau P_{pub}))$, $\beta = \sum_{j \in Q} v_j \cdot m_j \cdot R_j$, and the calculation cost is $2cH + (c-1)A_G + (5c+1)M_Z + (4c-4)A_Z + (c+4)M_G$. At the Verify stage, TPA calculates $\tau = H_2(PI\ D \| D)$, calculate $w_j = H_3(X \| R_j \| id_j)$, $\phi_j = H_4(D \| R_j \| id_j)$ for $j \in Q$, and equation (1) is also calculated. The calculation cost is $(2c+1)H + 4A_G + (3c+1)M_Z + (3c-3)A_Z + 4M_G$. The computational costs of the two schemes at each stage are compared as Table 2.

As we can see from Table 2, in the improved scheme, DO reduces the computational overhead of $2nH + 2nM_Z$ at the TagGen phase. At the ProofGen phase, CSP needs to bear the extra computation overhead of $2cH + (3c+1)M_Z + (3c-3)A_Z + 4M_G$. At the Verify phase, TPA needs to bear the extra computation overhead of $A_G + (c+1)M_Z + (c-1)A_Z + M_G$. Notice that the value of n is much larger than the value of c, the extra computing overhead borne by CSP and TPA is far less than the reduced computing overhead by DO, and the improved solution is more user-friendly and more efficient.

7. Conclusion

In this paper, we point out that Ming and Shi's scheme is insecure. The aggregated data blocks required for the audit are easy to forge. CSP can provide the correct integrity proof after modifying or deleting the data, and TPA will give the correct integrity audit results. In addition, to solve the idle tags problem in the existing audit schemes, we propose the idea of intermediate tags, which can save computing power for users. Finally, we apply the idea to the improved scheme and upgrade the original scheme on security to solve the security problems of the Ming and Shi's scheme and improve the audit efficiency. We hope that our idea of intermediate tags can be used by more scholars to construct more efficient audit solutions and the security issue pointed by us can be avoided when they design the scheme.

Data Availability

The datasets of this article are available on request from the authors.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Xiuguang Li and Ruifeng Li are responsible for the writing of the article and the construction of the improved scheme, Xu An Wang is responsible for the derivation of the formulas in the article and gives some significant ideas, Ke Niu is responsible for the verification of the security of this article, Xiaoyuan Yang is responsible for the polishing of the language of the article and the collection of the information related to this article, and Hui Li revised the finished manuscript.

Acknowledgments

This work was supported by National Key Research and Development Program of China (no. 2017YFB0802000); National Natural Science Foundation of China (nos. 62172436, 62102452, and 61732022); National Natural Science Foundation of China Key Program (U1836203); State Key Laboratory of Public Big Data (no. 2019BDKFJJ008); Engineering University of PAP's Funding for Scientific Research Innovation Team (no. KYTD201805); and Engineering University of PAP's Funding for Key Researcher (no. KYGG202011).

References

[1] L. Song, Y. Miao, J. Weng, K.-K. R. Choo, X. Liu, and R. H. Deng, "Privacy-Preserving threshold-based image retrieval in cloud-assisted internet of things," *IEEE Internet of Things Journal*, vol. 20229 pages, Article ID 3142933, 2022.

[2] 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*, pp. 598–609, Association for Computing Machinery, Alexandria, VA, USA, October 2007.

[3] Z. Ma, J. Ma, Y. Miao et al., "Lightweight privacy-preserving medical diagnosis in edge computing," in *Proceedings of the 2021 IEEE World Congress on Services (SERVICES)*, p. 9, IEEE, Chicago, IL, USA, September 2021.

[4] Y. Ming and W. Shi, "Efficient privacy-preserving certificateless provable data possession scheme for cloud storage," *IEEE Access*, vol. 7, Article ID 122091, 2019.

[5] 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.

[6] H. Jin, H. Jiang, and K. Zhou, "Dynamic and public auditing with fair arbitration for cloud data," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 680–693, 2018.

[7] H. Tian, Y. Chen, C. C. Chang et al., "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 701–714, 2017.

[8] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.

[9] W. Guo, H. Zhang, S. Qin et al., "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Generation Computer Systems*, vol. 95, pp. 309–322, 2019.

[10] G. Hou, J. Ma, C. Liang, and J. Li, "Efficient audit protocol supporting virtual nodes in cloud storage," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 5, pp. 1–14, 2020.

[11] R. Mishra, D. Ramesh, and D. R. Edla, "BB-tree based secure and dynamic public auditing convergence for cloud storage," *The Journal of Supercomputing*, vol. 77, no. 5, pp. 4917–4956, 2021.

[12] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: multiple-replica provable data possession," in *Proceedings of the 28th International Conference on Distributed Computing Systems*, pp. 411–420, IEEE, Beijing, China, June 2008.

[13] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MuR-DPA: top-down levelled multi-replica Merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2609–2622, 2015.

[14] W. Guo, S. Qin, F. Gao et al., "Dynamic proof of data possession and replication with tree sharing and batch verification in the cloud," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 1813–1824, 2022.

[15] Z. Yaling and S. Li, "Dynamic flexible multiple-replica provable data possession in cloud," in *Proceedings of the 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 291–294, IEEE, Chengdu, China, July 2020.

[16] Y. Qi, X. Tang, and Y. Huang, "Enabling efficient verification of dynamic data possession and batch updating in cloud storage," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 6, pp. 2429–2449, 2018.

[17] K. Deng, M. Xu, and S. Fu, "Outsourced data integrity auditing for efficient batch dynamic updates," *Communications in Computer and Information Science*, vol. 1149, pp. 325–339, 2020.

[18] Z. Ma, J. Ma, Y. Miao et al., "Verifiable data mining against malicious adversaries in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 953–964, 2022.

[19] X. Wang, J. Ma, Y. Miao, X. Liu, and R. Yang, "Privacy-Preserving diverse keyword search and online pre-diagnosis in cloud computing," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 710–723, 2022.

[20] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K.-K. R. Choo, "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 72–83, 2019.

[21] B. Wang, B. Li, and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 43–56, 2014.

[22] X. Wang, W. Jiao, H. Yang, L. Guo, X. Ye, and Y. Guo, "Algebraic signature based data possession checking method with cloud storage," in *Proceedings of the 11th International Conference on Prognostics and System Health Management*, pp. 11–16, IEEE, Jinan, China, October 2020.

[23] F. Li, J. Ma, Y. Miao et al., "Towards efficient verifiable boolean search over encrypted cloud data," *IEEE Transactions on Cloud Computing*, vol. 2021, Article ID 3118692, 1 page, 2021.

[24] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 159–170, 2015.

[25] J. Li, J. Wu, G. Jiang, and T. Srikanthan, "Blockchain-based public auditing for big data in cloud storage," *Information Processing & Management*, vol. 57, no. 6, Article ID 102382, 2020.

[26] H. Yang, R. Su, P. Huang et al., "PMAB: a public mutual audit blockchain for outsourced data in cloud storage," *Security and Communication Networks*, vol. 202111 pages, Article ID 9993855, 2021.

[27] X. Yang, X. Pei, M. Wang, T. Li, and C. Wang, "Multi-replica and multi-cloud data public audit scheme based on blockchain," *IEEE Access*, vol. 8, Article ID 144809, 2020.

[28] H. Wang, H. Qin, M. Zhao, X. Wei, H. Shen, and W. Susilo, "Blockchain-based fair payment smart contract for public

cloud storage auditing," *Information Sciences*, vol. 519, pp. 348–362, 2020.

[29] P. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi, and N. Kumar, "Blockchain data-based cloud data integrity protection mechanism," *Future Generation Computer Systems*, vol. 102, pp. 902–911, 2020.

[30] I. Zikratov, A. Kuzmin, V. Akimenko, V. Niculichev, and L. Yalansky, "Ensuring data integrity using blockchain technology," in *Proceedings of the 20th Conference of Open Innovations Association (FRUCT)*, pp. 534–539, IEEE, Saint-Petersburg, Russia, April 2017.

[31] W. Xu, D. Feng, and J. Liu, "Public verifiable proof of storage protocol from lattice assumption," in *Proceedings of the 2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment*, pp. 133–137, IEEE, Beijing, China, July 2012.

[32] H. Liu and W. Cao, "Public proof of cloud storage from lattice assumption," *Chinese Journal of Electronics*, vol. 23, no. 1, pp. 186–190, 2014.

[33] X. Zhang, C. Xu, and C. Jin, "Enabling identity-based cloud storage public auditing with quantum computers resistance," *International Journal of Electronic Security and Digital Forensics*, vol. 8, no. 1, pp. 82–98, 2016.

[34] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Information Sciences*, vol. 472, pp. 223–234, 2019.

[35] C. Sasikala and C. Shoba Bindu, "Certificateless remote data integrity checking using lattices in cloud storage," *Neural Computing & Applications*, vol. 31, no. 5, pp. 1513–1519, 2019.

[36] C. Lan, H. Li, and C. Wang, "Cryptanalysis of "Certificateless remote data integrity checking using lattices in cloud storage," in *Proceedings of the 10th International Conference on Information Science and Technology (ICIST)*, pp. 134–138, IEEE, London, UK, September 2020.