Hindawi
Security and Communication Networks
Volume 2022, Article ID 6800216, 9 pages
https://doi.org/10.1155/2022/6800216

Research Article

Improved Public Auditing System of Cloud Storage Based on BLS Signature

Ruifeng Li, Haibin Yang, Xu An Wang, Zhengge Yi, and Ke Niu

Chinese People's Armed Police Force Engineering University, Xi'an, China

Correspondence should be addressed to Xu An Wang; wangxazjd@163.com and Ke Niu; niuke@163.com

Received 19 January 2022; Accepted 5 May 2022; Published 29 May 2022

Academic Editor: Qi Li

Cloud storage and cloud computing technologies have developed rapidly for a long time, and many users outsource the storage burden of their data to the cloud to obtain more convenient cloud storage services. Allowing users to audit the private data's integrity has become an additional basic function of the cloud server when providing services. In 2021, based on the BLS signature and automatic blocker protocol, Jalil et al. proposed a secure and efficient cloud data auditing protocol. The protocol can realize public audit, batch audit, data update, and protect data privacy. Moreover, the automatic blocker protocol is used to realize the identity authentication of the auditor. The protocol is relatively novel, innovative, and has a larger use space. However, we found that their scheme had security problems. If the cloud server has thoughts of malicious attack, he can forge the proof that he holds users' data with stored labels and pass the audit. Referring to the original protocol and being inspired by them, we propose an improved audit protocol. The improved protocol solves the security problem and is more effective.

1. Introduction

Recently, advanced and innovative technologies represented by cloud computing and cloud storage have become increasingly mature. Cloud storage and cloud computing technologies have the characteristics of convenience, economy, and high scalability. Users can store the generated data in the platform and control their data remotely without purchasing and using local storage devices. Users are increasingly inclined to use cloud storage services to manipulate data more quickly and easily.

Cloud server providers centrally hold massive amounts of users' data, which are easily targeted by malicious attackers, and dishonest cloud server providers will deliberately delete users' data or conceal data security incidents from users for reasons such as reducing their own storage burden or maintaining their reputation. In the application of cloud storage technology, users cannot absolutely manipulate the data, and the integrity of the users' data is threatened. Verifying the integrity of cloud data is a hot topic of current research now.

1.1. Organization. We organize our paper as follows: in Section 1, we introduce the research background and related work. In Section 2, we describe the system model of cloud storage audit protocol. In Section 3, we review Jalil et al.'s public audit protocol. In Section 4, we give our attack on the original protocol and show that it is not efficient. In Section 5, we introduce our improved secure auditing protocol. In Section 6, we analyze the security of the improved protocol and compare the audit efficiency of the improved protocol with the original protocol. Finally, in Section 7, we make the conclusion of our work.

1.2. Related Work. Scholars have proposed many cloud storage data integrity audit protocols with different functions to meet the different needs of users in different application scenarios more effectively. In 2004, based on the RSA signature, Dewarte et al. [1] designed a protocol to audit remote files. However, the exponential calculation on all data blocks in the file will be performed on the user side, which will result in expensive computational overhead. In 2007,

Ateniese et al. [2] designed a verification scheme suitable for a cloud storage environment called "provable data possession (PDP)." The protocol uses an RSA-based homomorphic linear authenticator and random sampling technology, and users only download the partial file to be able to verify the integrity. Then, Juels and Kaliski [3] designed another "proof of retrievability(PoR)" scheme suitable for a cloud storage environment, which implements data integrity detection by inserting special data blocks (generally called "sentinels") into the data file.

In the actual application of cloud storage, users may need to perform various modifications and update operations on the data. Therefore, researchers have proposed audit protocols that support dynamic data updates. In 2008, Ateniese et al. [4] first proposed an audit protocol that can achieve dynamic data update with a symmetric encryption method. However, this audit protocol has the shortcoming of limited audit numbers and does not support public data audit. In 2012, Zhu et al. [5] constructed an audit protocol that supports dynamic data update with an index hash table (IHT) based on zero-knowledge proof. In 2015, Erway et al. [6] designed an audit protocol based on the sorted authentication skip list. The protocol supports a complete data dynamic update. In 2016, Jin et al. [7] introduced an index switcher to propose an audit protocol that not only provides fair arbitration but also supports dynamic data updates. In 2017, Shen et al. [8] used a two-way linked list of location arrays to implement the audit of the data. The protocol uses global and block-free sampling verification methods, which can also reduce computing and communication costs. In 2019, Guo et al. [9] designed a verification protocol that supports task outsourcing and supports dynamic data updates. It provides a log audit mechanism to enable users to detect misconduct by dishonest auditors. However, the solution has security loopholes. After multiple audits of data blocks with the same index, theoretically, data labels can be forged by solving linear independent equations. In 2020, the cloud audit scheme suitable for IoT [10] designed by Hou et al. [11] uses a chameleon authentication tree to save the computational overhead during the dynamic data update process and supports batch audit.

If users undertake the periodic audit work, it will generate a large computational overhead and consume a lot of resources [12]. In practical application scenarios, it is important to protect the privacy of user's data [13]. Scholars introduce a third-party auditor (TPA) to help users regularly check the integrity of the data stored on the cloud server. However, when users outsource the audit task, TPA will obtain data content during the implementation of audit tasks [14]. In 2013, Wang et al. [15] designed a public verification scheme, and the scheme supports a privacy protection function based on random masking technology and batch audit function based on the homomorphic linear authenticator. The protocol ensures that TPA cannot obtain the user's real data during the data integrity audit process. In 2014, Worku et al. [16] used random masking technology to propose an efficient public audit protocol with data privacy protection function. Wang et al. [17] designed a shared data audit protocol, which uses the ring signature technology and can protect the users' identity privacy. In 2015, Xiong et al. [18] used an ID-based encryption algorithm to design a privacy protection protocol, and the protocol uses distributed hash table network to protect sensitive data. In 2016, Li et al. [19] used online/offline signatures to design a lightweight public audit protocol with data privacy protection function.

Traditional cloud audit protocols are mostly based on the design of PKI cryptosystem, which brings complicated certificate management issues. In 2013, the first public identity-based audit scheme was designed by Zhao et al. [20]. The protocol minimizes the information carried in the verification process and the information obtained or stored by TPA, which simplifies key management and reduces communication and calculation overhead. In 2014, Wang et al. [21] proposed an ID-based data audit scheme, which formally defines the ID-based remote file verification model. The protocol gave the first security proof of the identity-based audit protocol based on CDH problem's difficulty. In 2016, Wang et al. [22] designed an agent-oriented ID-based remote data audit protocol. According to user's authorization, the protocol can realize three modes of private audit, entrusted audit, and public audit. In the same year, Yu et al. [23] used zero-knowledge proof to propose an ID-based cloud audit protocol that supports the privacy protection of users' data. The protocol regulates the identity-based audit protocol and its security model and can realize zero-knowledge privacy protection for TPA. In 2019, as the solution to the complex key management problem in cloud data integrity verification, Li et al. [24] used fuzzy identity to design an audit protocol. Xue et al. [25] designed an ID-based audit protocol using blockchain to construct random challenge messages. In their protocol, TPA cannot forge audit results to deceive users [26]. Peng et al. [27] designed a new ID-based data ownership verification protocol using compressed authentication arrays, which can simultaneously and efficiently support batch verification for multiple users in terms of computing and communication. Rabaninejad et al. [28] used the online/offline signature to design an ID-based PDP, and the protocol is implemented to support privacy protection, batch audit, and full dynamic data update [26].

However, the key escrow problem exists in ID-based cloud audit protocols, so many cloud audit protocols based on certificateless signature have been proposed. In the certificateless signature system, the user and the key generation center (KGC) cooperate to produce the private key for the user, which can avoid the strong dependence of the system security on the KGC security [29]. In 2013, Wang et al. [30] designed a certificateless cloud audit protocol, but He et al. [31] later pointed out the security problem. In 2015, Zhang et al. [32] designed the certificateless cloud data verification protocol that can resist malicious auditors. In 2017, Kang et al. [33] applied the certificateless cloud audit protocol to wireless body area networks. The proposed protocol can resist malicious auditors and protect data content. The certificateless cloud audit protocol proposed by He et al. [34] can protect users' privacy, but it has also been pointed out that there are security problems. He et al. [35]

applied the certificateless data audit protocol to the data management system of the smart grid, reducing the computational overhead. In 2018, Yang et al. [36] designed a certificateless cloud audit scheme for group user file sharing, which supports the protection of data content and users' identity privacy. In 2019, Wu et al. [37] defined the security model of the certificateless cloud audit protocol with privacy protection. The proposed protocol supports the protection of multiuser group identity privacy. In 2020, Huang et al. [38] designed a certificateless data verification protocol supporting the batch audit function, which realized efficient key update based on the Chinese remainder theorem.

1.3. Our Contribution. Recently, Jalil et al. [39] proposed an effective cloud data public audit protocol based on BLS signature to realize public audit and protect file content privacy. The protocol implements batch audit and dynamic update. Their scheme also uses automatic blocker protocol (ABP) to prevent unauthorized TPA from participating in the audit work, which is highly innovative, and ABP is essentially an access control facility [40], which can detect threats from auditors [41]. However, we found that their protocol has security issues. Even if the cloud server does not hold the stored data, he can mathematically prove that he holds the user's data. Then, we propose an improved and secure protocol with high security. The analysis shows the safety and effectiveness of our improved program in actual environments.

2. System Model

To facilitate understanding, we define and explain the various symbols and variables that appear in the original scheme and the improved scheme in Table 1.

The existing cloud audit systems generally include three interactive entities: cloud server provider (CSP) provides users with data storage services to obtain remuneration. CSPs are incredible. They may delete cloud data for profit or steal users' data privacy. Users: users are the owners of the data, and they upload files to the cloud to save their own storage cost. Third-party auditor (TPA) is not an entirely believable auditor entrusted by users, and on the one hand, TPA performs the audit task faithfully, and on the other hand, TPA attempts to decipher the content of the user's data with curiosity.

The interaction process of all entities: the user preprocesses the data to be stored and uploads it to CSP. When the data integrity needs to be verified, TPA generates a challenge with relevant parameters and sends them to CSP. Based on the challenge parameters, CSP uses cloud data to generate the proof that he holds the user data in full and sends the proof to TPA. TPA uses the proof to audit the data's integrity and sends the result to the user.

3. Review of Jalil et al.'s Protocol

There are three entities involved in Jalil et al.'s scheme. Jalil et al. used the BLS signature to achieve public audit and protect data content privacy. The program also supported batch audit and dynamic update. In addition, the proposed system enhanced the level of security authentication through an ABP to protect the system from unauthorized TPA. In particular, their scheme contains the following algorithm.

Table 1: Notations.

Notations	Descriptions
$(b_1, ..., b_n)$	Unencrypted n data blocks
$(e_1, ..., e_n)$	Encrypted n data blocks
G	Multiplicative cyclic group
E	Bilinear mapping
H	Secure hash function $H(\cdot)$: $\{0,1\}^* \longrightarrow Z_q^*$
Z_q^*	Prime field
g	Generator of G
λ	System initialization parameter
F	User's data file
k_s	User's secret key
m_i	Name of data block e_i
k_p	Public key of user
Q	Challenged subset of $(1, n)$
S_i	Authentication label for e_i
S	Collection of S_i
c	Number of challenged data blocks
a_i, r, p_i	Random values
V, μ, μ', R	Intermediate parameter
S_U	Collection of authentication labels of multiusers
$\lvert Z_q^* \rvert$	The size of an element of Z_q^*
$\lvert S \rvert$	The size of a label
$\lvert E_G \rvert$	The computational cost of a power on G
$\lvert M_G \rvert$	The computational cost of a multiplication on G
$\lvert A_G \rvert$	The computational cost of an add on G
$\lvert E \rvert$	The computational cost of a bilinear mapping
$\lvert H \rvert$	The computational cost of a hash

3.1. DataProtection Protocol. To protect data privacy, data file blocks need to be encrypted first. The user divides the data file F into n data blocks $(b_1, \cdots, b_n)$ and then uses the AES encryption algorithm to encrypt the data blocks and obtain the encrypted data blocks $(e_1, \cdots, e_n)$.

3.2. Setup Protocol. The user takes the security parameter $\lambda \in Z_q^*$ as input, for each data block e_i, outputs the corresponding private key $k_{s_i} \in Z_q^*$, and calculates the corresponding public key $k_{p_i} = g^{k_{s_i}} \in G$.

3.3. SignatureGen Protocol. For each data block e_i, the user generates a random value $a_i \in Z_q^*$ and calculates the corresponding label S_i:

$$S_i = \left(H\left(m_i\right) \cdot g^{a_i}\right)^{k_{s_i}}, \tag{1}$$

where m_i is the name of relevant blocks e_i and H is SHA256 hash function, which defines intermediate parameters $V_i = g^{a_i} \in G$.

Then, the user uploads V_i and m_i to the auditor and uploads e_i and S_i with pk_i to cloud for $i \in [1, n]$ and deletes the local data.

3.4. ChallGen Protocol. When the user needs to verify the integrity of cloud data, he sends an audit request to the TPA. TPA first randomly selects c elements to form a subset Q of $[1, n]$. For all $i \in Q$, TPA selects a random $p_i \in Z_q^*$ and sends all i and p_i to CSP.

3.5. Response Protocol. When CSP receives an audit challenge from TPA, he first asks the user whether the user has issued an audit request, thereby confirming the authenticity of the challenge from the TPA. After receiving user's affirmative reply, CSP confirms that the challenge is true and performs the next step. This process is implemented through the ABP. CSP uses the following equation to calculate the aggregate tag and sends the evidence S to the auditor:

$$S = \prod_{i=1}^{c} S_i. \tag{2}$$

3.6. CheckProof Protocol. When the TPA receives the corresponding evidence generated by the CSP for the challenge, he calculates the following equation to verify the integrity of the data:

$$E(S, g) = \prod_{i=1}^{c} E\left(H(m_i) \cdot V_i, k_{p_i}\right). \tag{3}$$

If equation (3) is true, he shows that the CSP has faithfully performed the service and ensured the integrity of the cloud data.

3.7. BatchAuditing Protocol. Each user divides the original file into n data blocks, then uses different encryption keys to encrypt the respective data blocks, generates private and public keys for different data blocks, and uses equation (1) to generate data tags. All users send $(e_i, S_i, pk_i, i\ d)$ for $i \in [1, n]$ to the cloud and upload metadata $(m_i, V_i, i\ d)$ to TPA, where $i\ d$ represents the user's identifier. When the data integrity needs to be verified, TPA randomly selects c data block indexes to be challenged and sends them to CSP. After CSP receives the challenge and confirms the authenticity of the challenge, based on the label set S_j of each user, the aggregate label S_U is calculated for all challenged data blocks:

$$S_U = \prod_{j=1}^{u} S_i, \quad \text{where}\,[1 \le j \le u]. \tag{4}$$

CSP generates evidence $(S_U, k_{p_{ij}})_{(1 \le i \le c, 1 \le j \le u)}$ and sends it to TPA. After receiving the evidence, the TPA verifies whether the following equation holds:

$$E(S_U, g) = \prod_{j=1}^{u} \left\{ E\left(\prod_{i=1}^{c} H(m_i)_j \cdot V_{ij}, k_{p_{ij}} \right) \right\}. \tag{5}$$

If equation (5) is true, it means that the integrity of the data has not been damaged.

4. Our Attack

In the audit protocol of Jalil et al.'s scheme, the correctness of the audit cannot be achieved. Even if the user's data held by the CSP are incomplete, CSP can pass the audit. In the SignatureGen protocol, the user calculates the signatures $(\{S_i\}_{[1 \le i \le n]})$ as equation (1). In equation (1), the calculation process of $(\{S_i\}_{[1 \le i \le n]})$ is determined by the private key value sk_i and the name of the data block m_i. However, $(\{S_i\}_{[1 \le i \le n]})$ are not signatures of the content e_i. In response protocol, CSP only uses equation (2) to calculate the aggregation signature, but he does not calculate the aggregation of the data content. The integrity proof generated by the CSP has nothing to do with the content of the data block. The CSP can use the stored signatures $(\{S_i\}_{[1 \le i \le n]})$ to generate the integrity evidence and pass the audit, so he can store the name m_i locally instead of the content e_i. In addition, in the original scheme, the number of public keys and private keys required is extremely large, which is proportional to n. Both in terms of certificate management and storage overhead of three entities, it is more complicated and cumbersome. In the CheckProof protocol, c bilinear mappings are used. The cost of calculation is also relatively high. In this section, we will show that CSP can generate an integrity proof that passes the audit from TPA without the store data block e_i.

The relevant data stored by CSP include the following:

$$\begin{aligned} &e_1, e_2, \ldots, e_n, \\ &S_1, S_2, \ldots, S_n, \\ &m_1, m_2, \ldots, m_n, \\ &k_{p_1}, k_{p_2}, \ldots, k_{p_n}. \end{aligned} \tag{6}$$

User needs to store the following:

$$\begin{aligned} &k_{s_1}, k_{s_2}, \ldots, k_{s_n}, \\ &k_{p_1}, k_{p_2}, \ldots, k_{p_n}. \end{aligned} \tag{7}$$

The data stored by TPA include the following:

$$\begin{aligned} &m_1, m_2, \ldots, m_n, \\ &V_1, V_2, \ldots, V_n, \\ &k_{p_1}, k_{p_2}, \ldots, k_{p_n}. \end{aligned} \tag{8}$$

We can see that the storage costs of the three entities are proportional to n, and the storage costs are relatively large, which violates the original intention of cloud storage. In addition, CSP and TPA need to store n public keys, users need to store the same number of private and public keys as the number of e_i requiring a lot of certificates, and certificate management is more complicated.

In the response protocol of Julil et al.'s protocol, CSP only generates the aggregation of signatures. CSP stores S_i, so regardless of whether CSP stores data, aggregate tags S can be generated according to equation (2). As long as the stored signatures are correct, the correct data audit proof can be generated and verified by the CSP.

In the CheckProof stage, after the auditor accepts the proof, he needs to verify whether equation (3) is true or not and calculates c bilinear mappings. The bilinear mapping is computationally expensive and reduces the audit efficiency.

5. Improvements to the Secure Auditing Protocol

Based on the above analysis, the original protocol is improved here to enhance security and efficiency. The difference comparison between the original scheme and the improved scheme is shown in Figure 1.

5.1. DataProtection Protocol. The user encrypts n data blocks $(b_1, \cdots, b_n)$ divided from the data file F using the AES encryption algorithm and obtains the encrypted data blocks $(e_1, \cdots, e_n)$, which can protect data privacy.

5.2. Setup Protocol. CSP inputs security parameters λ and outputs public parameters $\{G, g, E, H\}$. Among them, G is a multiplicative cyclic group, g is the generator of G, E is the bilinear mapping, and H is the hash function. The user randomly generates $k_s \in Z_q^*$ and calculates $k_p = g^{k_s} \in G$.

5.3. SignatureGen Protocol. For each data block e_i, the user calculates the corresponding label S_i:

$$S_i = \left(H(i) \cdot g^{e_i}\right)^{k_s}, \tag{9}$$

The tag $(\{S_i\}_{[1 \le i \le n]})$ is calculated by the secret key k_s, data block e_i, and data block index i. Then, the user deletes the local data and tags after uploading them to the cloud.

5.4. ChallGen Protocol. To verify whether the data are complete, the user sends a message to TPA requesting an audit first. TPA randomly selects c elements from $(1, n)$ to form a subset Q, and then, he randomly selects $p_i \in Z_q^*$ for all $i \in Q$. Finally, all i and p_i are sent to CSP.

5.5. Response Protocol. When CSP receives an audit challenge from TPA, he first ensures the authenticity of the challenge by querying the user. When the user's authenticity is confirmed, the CSP will accept the challenge. This process is implemented through the ABP. CSP randomly generates $r \in Z_q^*$ and uses the following equations to calculate the proof:

$$R = k_p^r, \tag{10}$$

$$S = \prod_{i=1}^{c} S_i^{p_i}, \tag{11}$$

$$\mu\prime = \sum_{i=1}^{c} p_i e_i, \tag{12}$$

$$\mu = \mu' + r, \tag{13}$$

and then sends the proof $\{R, S, \mu\}$ to the auditor.

5.6. CheckProof Protocol. When the CSP sends the evidence to the TPA, TPA verifies the authenticity of equation (14):

$$e(S \cdot R, g) \stackrel{?}{=} e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p\right). \tag{14}$$

If equation (14) is true, the data are completed and not corrupted. The process of proving the truth of equation (14) is as follows:

$$\begin{aligned} e(S \cdot R, g) &= e\left(\prod_{i=1}^{c} S_i^{p_i} \cdot \left(g^{k_s}\right)^r, g\right) = e\left(\prod_{i=1}^{c} \left(\left(H(i) \cdot g^{e_i}\right)\right)^{p_i} \cdot g^r, g^{k_s}\right) = e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot \prod_{i=1}^{c} g^{e_i p_i} \cdot g^r, k_p\right) \\ &= e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\sum_{i=1}^{c} e_i p_i + r}, k_p\right) = e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p\right). \end{aligned} \tag{15}$$

5.7. BatchAuditing Protocol. u users use different encryption keys to encrypt the data blocks belonging to themselves among the n data blocks divided from the original file, generate private keys $k_{s_j(1 \le j \le u)}$ and public keys $k_{p_j(1 \le j \le u)}$, and then use equation (9) to generate data tags. All users delete the local data after the task of transferring (e_i, S_i) to the cloud server is completed. To prove the completeness of the data, the TPA randomly selects k data block indexes to be challenged, sending the indexes and corresponding random values $p_{i(1 \le i \le c)}$ to the CSP. After the CSP receives and confirms the authenticity of the content, TPA randomly generates $r_j \in Z_q^*$ for each user and calculates:

$$R = \prod_{j=1}^{u} k_{p_j}^{r_j} = \prod_{j=1}^{u} g^{k_{s_j} \cdot r_j}, \tag{16}$$

$$\mu_j' = \sum_{i=1}^{c} p_i \cdot e_{ij}, \tag{17}$$

$$\mu_j = \mu_j' + r_j. \tag{18}$$

Based on the set $S_{j(1 \le j \le u)}$ of each user, the aggregate tag S_U is calculated for all challenged data blocks:

$$S_U = \prod_{j=1}^{u} \prod_{i=1}^{c} S_{ij}^{p_i}. \tag{19}$$

CSP generates evidence $P = (S_U, k_{p_{ij}})_{(1 \le i \le c, 1 \le j \le u)}$ and sends it to TPA as a basis for the verification. Upon receipt, TPA indicates whether the cloud data are completed by verifying the following equation:

$$E(S_U \cdot R, g) \stackrel{?}{=} \prod_{j=1}^{u} E\left(\left(\prod_{i=1}^{c} H(i)^{q_i} \cdot g^{\mu_j}, k_{p_j}\right)\right). \tag{20}$$

If equation (20) holds, it proves that data integrity has not been compromised. The proof of the correctness of (15) is as follows:

$$\begin{aligned} & E(S_U \cdot R, g) \\ &= E\left(\prod_{j=1}^{u} \prod_{i=1}^{c} S_{ij}^{p_i} \cdot \prod_{j=1}^{u} g^{k_{s_j} \cdot r_j}, g\right) \\ &= \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} \left(H(i) \cdot g^{e_{ij}}\right)^{k_{s_j} p_i} \cdot g^{k_{s_j} \cdot r_j}, g\right) \\ &= \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} \left(H(i) \cdot g^{e_{ij}}\right)^{p_i} \cdot g^{r_j}, g^{k_{s_j}}\right) \\ &= \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} H(i)^{p_i} g^{\sum_{i=1}^{c} e_{ij} p_i + r_j}, k_{p_j}\right) \\ &= \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} H(i)^{p_i} g^{\mu}, k_{p_j}\right). \end{aligned} \tag{21}$$

6. Analysis of the Improved Protocol

The security of the improved protocol is first analyzed and explained here, including preventing forgery attack from CSP and attack from TPA to steal data content privacy. Then, the storage and computation overhead of the improved protocol are analyzed in comparison with the original protocol, to prove that the improved protocol is safe and efficient.

6.1. Security Analysis.

(1) Anti-Forgery Attack: if in the cloud, the CSP generates a forged audit certificate $\tilde{\mu}$ and the stored user data are corrupted or tampered with, then it means that the group can compute the discrete logarithm problem with probability $1 - 1/q$ (q is a large prime number). A forged data possession proof $\tilde{\mu} = \sum_{i=1}^{c} p_i \tilde{e}_i + r$ will be generated by the CSP in the case of incorrect data, and we define the following:

$$\begin{aligned} \Delta\mu &= \tilde{\mu} - \mu \\ &= \tilde{\mu}' - \mu' \\ &= \sum_{i=1}^{c} p_i \tilde{e}_i - \sum_{i=1}^{c} p_i e_i \\ &= \sum_{i=1}^{c} p_i \Delta e_i. \end{aligned} \tag{22}$$

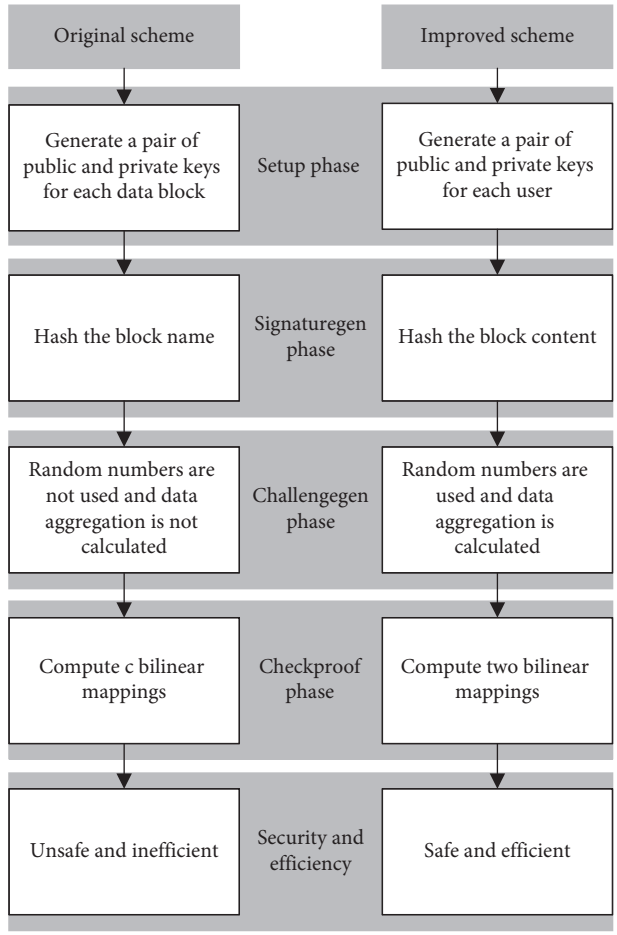

Figure 1: Difference between two schemes.

Because $\tilde{\mu}$ is the forged evidence, there must be a difference between $\tilde{\mu}$ and μ, and there is at least one $\Delta e_i \neq 0$. Assuming that CSP's forged proof of data possession $\tilde{\mu}$ can pass TPA's audit, therefore

$$E(S \cdot R, g) = E\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\tilde{\mu}}, k_p\right). \tag{23}$$

The correct proof can pass the TPA audit; therefore,

$$E(S \cdot R, g) = E\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p\right). \tag{24}$$

From equations (23) and (24), we get $E(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p) = E(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\tilde{\mu}}, k_p)$, so $g^{\mu'} = g^{\tilde{\mu}'}$ and $g^{\Delta\mu} = 1$. Since G is a cyclic group, so $\forall a, b \in G$, $\exists x \in Z_q^*$ makes $b = a^x$.

Given a and b, then g can be written as $g = a^{y_1} \cdot b^{y_2} \in G$, and $y_1, y_2 \in Z_q^*$, and therefore,

$$1 = g^{\Delta\mu} = \left(a^{y_1} b^{y_2}\right)^{\Delta\mu} = a^{y_1 \Delta\mu} b^{y_2 \Delta\mu}. \tag{25}$$

simplified further to get $b = a^{-y_1 \Delta\mu / y_2 \Delta\mu}$.

To make equation (25) not true, only if the denominator $y_2 = 0$, then equation (25) is meaningless,

Table 2: Storage cost comparison.

	CSP	TPA	USER
Original protocol	$n\|Z_q^*\| + n\|S\|$	$3n\|Z_q^*\|$	$2n\|Z_q^*\|$
Improved protocol	$n\|Z_q^*\| + n\|S\|$	$\|Z_q^*\|$	$2\|Z_q^*\|$

Table 3: Calculation cost comparison.

	CSP	TPA	USER
Original protocol	$c\|M_G\|$	$2c\|M_G\| + c(E) + c(H)$	$4n\|E_G\| + n\|H\| + n\|M_G\|$
Improved protocol	$c\|M_G\| + c\|E_G\|$	$c\|M_G\| + 2\|E\| + c\|H\| + c\|E_G\|$	$n\|E_G\| + n\|H\| + n\|M_G\|$

and $y_1, y_2 \in Z_q^*$, so $P[y_2 = 0] = 1/q$, and the probability that equation (25) is true is $1 - 1/q$.

It is concluded that if the CSP can successfully forge the data block, then he can calculate the discrete logarithm problem and the probability is $1 - 1/q$ but obviously the discrete logarithm problem is a difficult problem, so the CSP cannot forge the fake data block that has passed the audit.

(2) Privacy Protection: first, an authentication protocol (ABP) is used to prevent unauthorized adversaries from entering the system.

Then, in the DataProtection protocol, the user's original data $(b_1, \cdots, b_n)$ are encrypted by AES to obtain $(e_1, \cdots, e_n)$. The data uploaded to the cloud are encrypted data. The CSP does not hold the encryption and decryption keys of the AES encryption algorithm, so it is impossible to know the real data content of the user, avoiding the leakage of data privacy.

Finally, for TPA, the improved protocol uses random masking technology to realize data protection. Assuming that TPA is curious about the challenged data blocks' content $(e_1, \cdots, e_c)$ and audits c data blocks $t\,(t \geq 1)$ times, q_{ji} is represented as random parameters during the j – th time audit on the i – th data block, and then, the set of random numbers is $Q = \{p_{ij}\}_{1 \leq i \leq c, 1 \leq j \leq t}$. An evidence set consisting of t pieces is proofs $= \{(\mu_j, S_j, R_j)\}_{1 \leq j \leq t}$. TPA can obtain the following equations:

$$\begin{cases} p_{11}e_1 + p_{12}e_2 + \cdots + p_{1c}e_c + r_1 = \mu_1 \\ p_{21}e_1 + p_{22}e_2 + \cdots + p_{2c}e_c + r_2 = \mu_2 \\ \vdots \\ p_{t1}e_1 + p_{t2}e_2 + \cdots + p_{tc}e_c + r_t = \mu_t. \end{cases} \tag{26}$$

In the above equations, TPA knows $\{p_{ij}\}_{1 \leq i \leq c, 1 \leq j \leq t}$ and $\{\mu_j\}_{1 \leq j \leq t}$, but he does not know $\{e_i\}_{1 \leq i \leq c}$ and $\{r_j\}_{1 \leq j \leq t}$. There are $c + t$ unknown numbers in equation (26), no matter how many times TPA audits the same data blocks; that is, no matter what the value t is, it will always be less than $c + t$, and TPA cannot solve equation (26) and cannot know the content of the data blocks $(e_1, \cdots, e_c)$ and $(b_1, \cdots, b_c)$.

6.2. Efficiency Analysis. In the original protocol, the user needs to generate the corresponding public keys k_{p_i} and private keys k_{s_i} for $e_{i(1 \leq i \leq n)}$. After uploading the data blocks and tags $(\{S_i\}_{[1 \leq i \leq n]})$ to CSP, the user still needs to store his own public keys and private keys, and the storage cost is $2n|Z_q^*|$. In addition to data blocks, CSP also needs to store tags, and the storage cost is $2n|Z_q^*|$. $(\{v_i, m_i, pk_i\}_{[1 \leq i \leq n]})$ is stored at TPA, so the storage overhead is $3n|Z_q^*|$.

In the improved protocol, the user only holds a pair of k_p and k_s, and the storage overhead on the user side is $2|Z_q^*|$. CSP needs to store $(\{S_i, e_i\}_{[1 \leq i \leq n]})$, and the storage overhead is $n|Z_q^*| + n|S|$. When TPA verifies the evidence, he needs the user's public key in addition to the challenge information, and the storage cost is $|Z_q^*|$. The storage cost comparison between the original protocol and the improved protocol is shown in Table 2. The storage overhead of the improved scheme is lower than that of the original scheme.

Because the multiplication and addition operations on Z_q^* have minimal computational overhead compared with other operations, we omit them. In the original protocol, the user needs to calculate $k_{p_i} = g^{k_{s_i}}$, $S_i = (H(m_i) \cdot g^{a_i})^{k_{s_i}}$, and $V_i = g^{a_i} \in G$, and the calculation cost is $4n|E_G| + n|H| + n|M_G|$. CSP needs to calculate $S = \prod_{i=1}^{c} S_i$, and the calculation cost is cM_G. TPA needs to calculate equation (3), and the calculation cost is $2c|M_G| + c|E| + c|H|$.

In the improved protocol, the user needs to calculate $S_i = (H(i) \cdot g^{e_i})^{k_s}$, and the calculation cost is $n|E_G| + n|H| + n|M_G|$. CSP needs to calculate $S = \prod_{i=1}^{c} S_i^{p_i}$ and $\mu\prime = \sum_{i=1}^{c} p_i e_i$, and the calculation cost is $c|M_G| + c|E_G|$. TPA needs to calculate equation (14), and the calculation cost is $c|E_G| + 2|E| + c(H) + c(M_G)$. The calculation cost comparison between the original protocol and the improved protocol is shown in Table 3. Among the entities of the improved scheme, only CSP's calculation overhead is slightly higher than the original scheme. The calculation overhead of TPA and user in the improved scheme is significantly reduced compared with the original scheme.

7. Conclusion

According to the analysis in this study, it is clear that the protocol of Jalil et al. is insecure. We point out the security loophole in the original protocol and attacked it, and then, we propose an audit scheme with higher

security and efficiency based on the directions that can be improved.

Data Availability

The data supporting this systematic review were taken from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Conflicts of Interest

There are no potential conflicts of interest.

Authors' Contributions

Ruifeng Li is 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, Haibin Yang is responsible for the polishing of the language of the article and the collecting of the information related to this article, Zhengge Yi is responsible for the verification of the security of this article, and Ke Niu revised the finished manuscript.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2017YFB0802000), National Natural Science Foundation of China (No. 62172436 and No. 62102452), 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] S. Jajodia and L. Strous, *Integrity and Internal Control in Information Systems VI*, IICIS, Lausanne, CA, USA, 2003.

[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, Alexandria, VA, USA, Octobner 2007.

[3] A. Juels and B. Kaliski, "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 584–597, Alexandria, VA, USA, Octobner 2007.

[4] G. Ateniese, R. Pietro, L. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, pp. 1–10, Istanbul, Turkey, September 2008.

[5] Y. Zhu, H. Hu, G. 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.

[6] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security*, vol. 17, no. 4, pp. 1–29, 2015.

[7] 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, 2016.

[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] K. Yu, Z. Guo, Y. Shen, W. Wang, J. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2698–2707, 2021.

[11] 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.

[12] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote E-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.

[13] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, "Achieving graph clustering privacy preservation based on structure entropy in social IoT," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 1–17, 2021.

[14] T. Deng, X. Li, J. Xiong, and Y. Wu, "POISIDD: privacy-preserving outsourced image sharing scheme with illegal distributor detection in cloud computing," *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 3693–3714, 2021.

[15] C. Wang, S. 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.

[16] S. Worku, C. Xu, J. Zhao, and X. He, "Secure and efficient privacy-preserving public auditing scheme for cloud storage," *Computers & Electrical Engineering*, vol. 40, no. 5, pp. 1703–1713, 2014.

[17] 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.

[18] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. Chen, "A full lifecycle privacy protection scheme for sensitive data in cloud computing," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1025–1037, 2015.

[19] J. Li, L. Zhang, J. K. Liu, H. Qian, and Z. Dong, "Privacy-preserving public auditing protocol for low-performance end devices in cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2572–2583, 2016.

[20] J. Zhao, C. Xu, F. Li, and W. Zhang, "Identity-based public verification with privacy-preserving for data storage security in cloud computing," *IEICE - Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E96.A, no. 12, pp. 2709–2716, 2013.

[21] H. Wang, Q. Wu, B. Qin, and J. Doming, "Identity-based remote data possession checking in public clouds," *IET Information Security*, vol. 8, no. 2, pp. 114–121, 2014.

[22] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1165–1176, 2016.

[23] Y. Yu, M. Au, G. Ateniese et al., "Identity-based remote data integrity checking with perfect data privacy preserving for

cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 767–778, 2017.

[24] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K. 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.

[25] J. Xue, C. Xu, J. Zhao, and J. Ma, "Identity-based public auditing for cloud storage systems against malicious auditors via blockchain," *Science China (Information Sciences)*, vol. 62, no. 3, pp. 45–60, 2019.

[26] L. Tan, K. Yu, C. Yang, K. Choo, and A. Bashir, "A blockchain-based Shamir's threshold cryptography for data protection in industrial internet of things of smart city," in *Proceedings of the 1st Workshop on Artificial Intelligence and Blockchain Technologies for Smart Cities with 6G*, pp. 13–18, New Orleans, Louisiana, United States, October 2021.

[27] S. Peng, F. Zhou, J. Li, Q. Wang, and Z. Xu, "Efficient, dynamic and identity-based remote data integrity checking for multiple replicas," *Journal of Network and Computer Applications*, vol. 134, no. 5, pp. 72–88, 2019.

[28] R. Rabaninejad, M. Asaar, M. Attari, and M. Aref, "An identity-based online/offline secure cloud storage auditing scheme," *Cluster Computing*, vol. 23, no. 5, pp. 1455–1468, 2019.

[29] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," in *Proceedings of the 9th International Conference on the Theory and Application of Cryptology*, pp. 452–473, Tainan, Taiwan, China, 2003.

[30] B. Wang, B. Li, H. Li, and F. Li, "Certificateless public auditing for data integrity in the cloud," in *Proceedings of the Communications and Network Security (CNS)*, pp. 136–144, Washington, D. C., USA, October 2013.

[31] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[32] 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.

[33] B. Kang, J. Wang, and D. Shao, "Certificateless public auditing with privacy preserving for cloud-assisted wireless body area networks," *Mobile Information Systems*, vol. 2017, no. 3, pp. 1–5, 2017.

[34] D. He, N. Kumar, H. Wang, L. Wang, and K. Choo, "Privacy-preserving certificateless provable data possession scheme for big data storage on cloud," *Applied Mathematics and Computation*, vol. 314, no. 1, pp. 31–43, 2017.

[35] D. He, N. Kumar, S. Zeadally, and H. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1232–1241, 2017.

[36] H. Yang, S. Jiang, W. Shen, and Z. Lei, "Certificateless provable group shared data possession with comprehensive privacy preservation for cloud storage," *Future Internet*, vol. 10, no. 6, pp. 1–17, 2018.

[37] G. Wu, Y. Mu, W. Susilo, F. Guo, and F. Zhang, "Privacy-Preserving certificateless cloud auditing with multiple users," *Wireless Personal Communications*, vol. 106, no. 3, pp. 1161–1182, 2019.

[38] L. Huang, J. Zhou, G. Zhang, and M. Zhang, "Certificateless public verification for data storage and sharing in the cloud," *Chinese Journal of Electronics*, vol. 29, no. 4, pp. 639–647, 2020.

[39] B. Jalil, T. Hasan, G. Mahmood, and H. Noman, "A secure and efficient public auditing system of cloud storage based on BLS signature and automatic blocker protocol," *Journal of King Saud University - Computer and Information Sciences*, vol. 2021, no. 9, pp. 1–14, 2021.

[40] Q. Li, B. Xia, H. Huang, Y. Zhang, and T. Zhang, "TRAC: traceable and revocable access control scheme for mHealth in 5G-enabled IioT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3437–3448, 2021.

[41] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: deep-learning-based threat detection in IioT," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 76–82, 2021.