Hindawi
Security and Communication Networks
Volume 2021, Article ID 6623639, 10 pages
https://doi.org/10.1155/2021/6623639

Research Article

A Blockchain-Based Public Auditing Protocol with Self-Certified Public Keys for Cloud Data

Hongtao Li,[1] **Feng Guo**,[2] **Lili Wang**,[3] **Jie Wang**,[1] **Bo Wang**,[1] **and Chuankun Wu**[2]

[1]*College of Mathematics & Computer Science, Shanxi Normal University, Linfen 041000, China*
[2]*School of Information Science and Engineering, Linyi University, Linyi 276002, China*
[3]*College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China*

Correspondence should be addressed to Feng Guo; 25576152@qq.com

Received 11 December 2020; Revised 18 January 2021; Accepted 11 February 2021; Published 23 February 2021

Academic Editor: Yinghui Zhang

Copyright © 2021 Hongtao 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 can provide a way to effectively store and manage big data. However, due to the separation of data ownership and management, it is difficult for users to check the integrity of data in a traditional way, which leads to the introduction of the auditing techniques. This paper proposes a public auditing protocol with a self-certified public key system using blockchain technology. The user's operational information and metadata information of the file are formed to a block after verified by the checked nodes and then to be put into the blockchain. The chain structure of the block ensures the security of auditing data source. The security analysis shows that attackers can neither derive user's secret key nor derive users' data from the collected auditing information in the presented scheme. Furthermore, it can effectively resist against not only the signature forging attacks but also the proof forging attacks. Compared with other public auditing schemes, our scheme based on the self-certified public key system has been improved in storage overhead, communication bandwidth, and verification efficiency.

1. Introduction

Cloud storage, which provides a way to effectively store and manage big data [1], is an important branch of cloud computing. Because cloud storage has superiorities of low cost, scalable, location-independent, and high performance [2–4], more and more individuals and businesses tend to outsource their data to the cloud. Although the advantages of cloud storage services are many and huge, it still faces a variety of security challenges [4–14].

For example, the security of data sharing and storage in the same group is an urgent issue to be solved in the cloud environment [6]. In other words, since the cloud users lose the management of data, a cloud service provider (CSP) must satisfy users' need for the security of stored data [7]. And users cannot verify the integrity of their data with traditional methods owing to the trust gap between users and CSP. In addition, cloud storage also faces many internal and external security threats [8–10]. Firstly, malicious attackers might do their best to retrieve users' outsourced data, even to destroy and delete the outsourced data. Then, the confidentiality, integrity, and availability of users' stored data are destroyed. Secondly, the user's outsourced data might also be illegally manipulated by CSP. For instance, CSP may selectively conceal certain errors in user's outsourced data due to Byzantine failures [11]. Furthermore, CSP might deliberately delete data that are rarely accessed by ordinary users in order to reduce storage space and save bandwidth [12, 13]. Finally, users may not be able to timely know the data changes and they may lack trust on CSP. Then, disputes arise, although those disputes may be caused by users' own improper operations [14]. Therefore, it is critical and significant to develop efficient data auditing techniques to check the confidentiality, integrity, and availability of stored data.

After data are outsourced to the cloud, users would delete local data and lose the management of outsourced data. Therefore, users can use audit technology to remotely verify whether the outsourced data are correct. The most core challenge of cloud data auditing is how to efficiently

check the cloud data integrity. To address this problem, a proof of retrievability (PoR) protocol [15] and a provable data possession (PDP) protocol [16] have been presented in 2007, respectively.

In typical PoR protocol, the user first encodes the data file with error-correcting code before outsourcing data to the CSP. Therefore, the user can reconstruct the entire file from the CSP's partial response. However, PoR protocol is applicable for static data. And it does not support third-party auditing and is a typical private auditing scheme. In private auditing, remote verification operation is performed directly between user and CSP. The user is the only source of verification results, while CSP and users do not trust each other and users cannot provide convincing auditing results for verification. Furthermore, user's burden is increased due to insufficient computing resources. Since one of the important motivations of outsourcing data is to reduce the user's burden of storage management, it is not recommended that users audit their data frequently.

To address this problem, a PDP protocol was first provided by Ateniese et al. [16]. In PDP, the RSA-based homomorphic authenticator is employed to check the data integrity and an independent authorized third-party auditor (TPA) was introduced. TPA can not only provide independent audit results but also bear the communication overhead and computation costs. Compared with PoR, PDP makes the process of verification more convenient and efficient and is more suitable for public auditing [5, 7, 10, 11, 16–25].

The public audit has advantages over private auditing, so it has attracted much attention of researchers. Since the idea of public auditing was raised in 2007 [16], a lot of auditing protocols have been designed in recent years [10, 12, 18–28].

In 2010, Wang et al. [22] also provided a similar architecture for public audit scheme with privacy-preserving property. To overcome the data leakage to the TPA, CSP integrates the aggregate value of the data blocks with random masking. However, the lack of strict performance analysis has greatly affected the practical application of the scheme. Furthermore, the length of data block must be equal to the size of cryptosystem. That means the storage space of tags generated for data blocks must be equal to the size of the original file [26]. This shows that the efficiency of the presented public auditing scheme is low. In order to improve the efficiency, Wang et al. [10] extended the above auditing protocol to multiuser settings. The extended protocol can support batch verification. However, the expected goal has not been achieved because the implementation of verification and updation brings higher computing and communication costs to TPA [27]. In 2011, Wang et al. [12] implemented complete data dynamics by using a Merkle hash tree (MHT), while the implementation of verification and updation also makes communication cost of protocol higher [21].

In 2013, Wang et al. [10] found that there was a risk of leakage of data information in the proposed scheme with public auditability [16]. Then, they designed a privacy-preserving scheme, which combined homomorphic linear authenticator (HLA) and random masking technique. Nevertheless, the designed scheme does not have the ability to protect the identity privacy of signers [28]. In order to reduce the computational cost and communication overhead, Zhu et al. [18] proposed a new public audit scheme based on index hash table (IHT), which is employed to organize the data properties for auditing. However, the index table is a sequence table. If you need to locate a certain element, it will take an average of half the total length of the table. This resulted in very efficient update operations, such as insertion and deletion [21]. In addition, these update operations would inevitably change the serial numbers of some blocks. Then, it is necessary to recalculate the tags of those blocks. In this way, CSP would require more extra computational costs and unnecessary communication overhead [19].

Then, Tian et al. [19] designed a public auditing protocol based on dynamic hash table (DHT) to support data dynamics, which claims to address the problem in Zhu's scheme [18]. The dynamic hash table is a single linked sequence table. Though the proposed public auditing protocol is efficient, there are still some drawbacks in this scheme. Firstly, because time stamps for verification are generated by the user and TPA only serves the user, CSP may suffer from the collusion attack launched by the user and TPA [21]. Secondly, there is no index switcher in the proposed scheme. Then, the relationship between the index number and the serial number of a certain data block cannot be clearly known. Finally, the proposed protocol still has relatively high computational costs.

In addition, Shen et al. [21] designed a novel public auditing protocol based on a new dynamic structure to overcome the drawbacks in [18]. The proposed dynamic structure consists of a doubly linked info table and a location array. Though the above protocols can effectively achieve public auditing, search operations in those schemes are relatively inefficient in the verification phase and the updating phase [27].

In 2018, Jin et al. [20] presented a scheme by employing an index switcher. Then, the relationship between the index number and the tag number of a certain data block can be clearly known. And there is no need to recalculate the tags caused by block update operations. Nevertheless, the index switcher needs to be periodically transformed among the systems, which will inevitably result in huge extra costs. Moreover, such an index switcher is not a complete structure. And how to switch between the two constituent tables is not explained in the proposed scheme [21].

In 2019, Ding et al. [29] proposed a public auditing protocol that is intrusion-resilient to mitigate the damage caused by key exposure problems. The protocol divides the lifetime of files stored in the cloud into several periods, each of which is further divided into several refreshing periods. The auditing key is updated every time period, and the secret value used to update the auditing key changes during each refreshing period. These two update operations are performed by the client and the third-party auditor (TPA).

In 2020, Garg et al. [30] proposed an efficient data integrity auditing method for cloud computing. The objective of this protocol is to minimize the computational complexity of the client during the system setup phase. Based on the

properties of bilinear pairings, the protocol is publicly verifiable and supports dynamic manipulation of data. The security of the protocol depends on the stability of the calculation of the Diffie–Hellman problem (CDHP) in the random oracle model (ROM).

The nature of blockchain is particularly suitable for data accounting and auditing. Because of its shared and fault-tolerant database, it has attracted the interest of the research community. Blockchain uses cryptography to build trust in peers to protect interactions of them. Meanwhile, it adopts consensus algorithm to ensure the block data are not changed, which is very suitable for data security in the cloud. In the past few years, some cloud security schemes based on blockchain have been proposed. Li et al. proposed a security framework for cloud data audit using blockchain technology, in which user's operational information on the file is formed to a block after validated by all checked nodes in the blockchain network and then put into the blockchain [31]. Linn et al. proposed a data auditing framework for health scenarios based on blockchain, in which blockchain was used as an access moderator to control the access to outsourced shared data [32]. Fu et al. introduced a privacy-aware blockchain-based auditing system for shared data in cloud applications [33]. Ghoshal et al. proposed an auditing mechanism based on blockchain structure, in which any user can perform the validation of selected files efficiently [34]. Fu proposed a blockchain-based secure data-sharing protocol under decentralized storage architecture [35]. Miao et al. proposed a decentralized and privacy-preserving public auditing scheme based on blockchain (DBPA), in which a blockchain is utilized as an unpredictable source for the generation of (random) challenge information, and the auditor is required to record the audit process onto the blockchain [36]. Li et al. proposed a public auditing scheme with the blockchain technology to resist the malicious auditors [37]. In addition, through the experimental analysis, we demonstrate that our scheme is feasible and efficient. Due to the limited capacity of blocks in the blockchain, only very important security information is considered to be stored in blocks; otherwise, the system performance will not be acceptable.

This paper proposes a public auditing protocol with a self-certified public key system using blockchain technology. The chain structure of the block ensures the security of auditing data source. Taking the security and efficiency into account, a novel public auditing scheme for cloud data is proposed in this paper based on a self-certified public key system. The contributions of this paper are as follows. Firstly, recent related public auditing protocol are introduced. Secondly, we propose a public auditing protocol with a self-certified public key system using blockchain technology, in which the security and efficiency are taken into account. Finally, we conduct detailed theory analysis of the security and efficiency of the new scheme.

The outline of the paper is as follows: the research background and necessary preliminaries for the new public auditing system are firstly introduced. In the latter, the corresponding algorithm of the proposed scheme is described. Then, the security and efficiency of the new scheme are comprehensively analyzed from four aspects. Finally, a few concluding remarks are given in the last section.

2. System Model and Desired Objectives

In general, our public auditing scheme includes the following four entities: *CSP*, *TPA*, *user*, and blockchain. The system model is shown in Figure 1.

CSP, who has large-scale computing and storage resources, provides users with on-demand data storage services. CSP is considered as an untrustworthy party. For their own self-interest or maintaining their reputations, CSP may choose to conceal the data errors from the users. To reduce the amount of storage space and save bandwidth, CSP may deliberately delete some data that users rarely access. Furthermore, the CSP may launch some attacks on TPA. For example, CSP may try to forge some legitimate data blocks and their corresponding tags in order to pass verification phase.

TPA, who undertakes audit tasks for users, provides fair and objective audit results. TPA is supposed to be credible but curious. More concretely, TPA can perform auditing credibly in the verification phase, but it may be curious about the privacy information of users' data and even may try to derive the users' data contents.

User, who has large amounts of data, outsources the data to the cloud. Then, he/she can enjoy the reliability of data storage and high-performance services. The maintenance overhead can also be reduced. However, due to the loss of the management of outsourced data, users will have a strong desire to periodically check the integrity and correctness of those data.

We use *blockchain* to store user's operations on the file and metadata information of the uploaded file. The system does not care where files are stored but only stores a file URL in metadata file. We take advantage of the blockchain's tamper-resistant nature to ensure the reliability of operation logs and file metadata. Metadata information is used to audit the integrity of the data, and the analysis operation log can be used for behavioral audit.

Based on the above description of public auditing scheme model, the desired objectives to be achieved must be given for designing a secure and efficient public auditing scheme.

Public Auditing. Any authorized TPA is allowed to verify the correctness and integrity of user's data stored in the cloud.

Blockless Verification. During the verification process, TPA does not need to audit cloud data by retrieving the data blocks.

Storage Correctness. CSP, who does not store the intact data as required, cannot pass the audit.

Privacy Preserving. TPA cannot derive users' data contents from the collected auditing information during the verification phase.

Batch Auditing. TPA can efficiently deal with multiple audit tasks from different users. It not only reduces the number of communications between TPA and CSP during the auditing phase but also enhances the verification efficiency [17, 23].

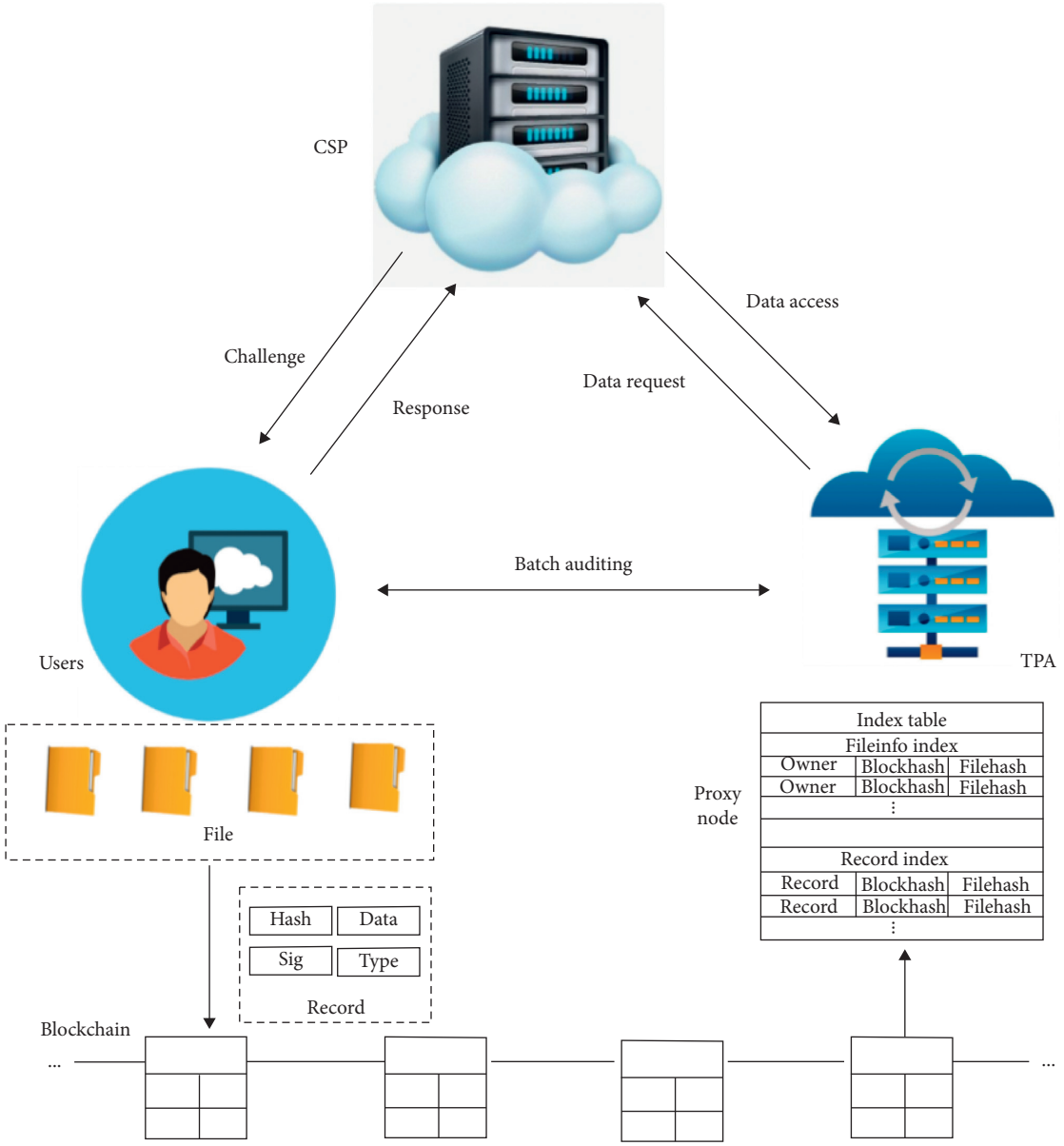

Figure 1: System model.

Lightweight. The public auditing scheme should have less communication overhead and lower computation cost.

3. Preliminaries

3.1. Self-Certified Public Key. The notion of self-certified public key (SCPK) was first introduced by Girault [38]. The user's public key is derived from the signature of the user's secret key with his/her identity in the SCPK system. The signature is signed by the system authority using the system's secret key. And the user's identity, public key, and secret key satisfy a computationally unforgettable mathematical relationship. While using the keys to perform encryption and decryption, signature verification, key agreement, or other cryptographic operations, the public key can be implicitly authenticated in the process of signature verification. In addition, each public key does not have a separate certificate and the verifier does not need to authenticate the certificate of the public key. Consequently, the SCPK system can reduce the storage space and computational overhead in public key schemes. Moreover, the user's private key is chosen by himself and the system authority who cannot get the private key from the transmitted data and cannot forge the signature as a user. Compared with ID-based public key system, the SCPK system has higher security and is more suitable for applications in open network environment.

3.2. Bilinear Map. Let G_1 and G_2 be two multiplicative cyclic groups of prime order p and g be a generator of G_1. A bilinear map is a map e: $G_1 \times G_1 \longrightarrow G_2$ with the following properties [39]:

Bilinearity: for all $u, v \in G_1$ and $a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$.

Computability: the map e is efficiently computable.

Nondegeneracy: $e(g, g) \neq 1$.

3.3. HVA. Homomorphic verifiable authenticator (HVA) is a basic component of public auditing [10, 19, 40]. Specifically, HVA can be generated based on digital signatures, such as RSA-based signature and BLS-based signature. Therefore, such HVAs can be considered as homomorphic verifiable signatures. Taking advantage of HVA, a public auditor can verify the integrity of outsourced data without downloading the original data. Generally speaking, HVA has the following properties [41, 42]:

Blockless Verifiability. Without knowing the actual data content, TPA can verify the integrity of the data blocks based on the proof constructed by HVAs.

Homomorphism. Let G_1 and G_2 be multiplicative groups, whose orders are a large prime p. Let "$\oplus$" and "$\otimes$" be operations in G_1 and G_2. If a map function $f: G_1 \longrightarrow G_2$ satisfies homomorphism, then $\forall g_1, g_2 \in G$, $f(g_1 \oplus g_2) = f(g_1) \otimes f(g_2)$.

Nonmalleability. Let σ_1 and σ_2 be signatures of block m_1 and block m_2, respectively. Given a certain block $m' = \alpha_1 m_1 + \alpha_2 m_2$, where α_1 and α_2 are two random numbers in $\mathbb{Z}_p^*$. For any user, if he/she does not know the private key, he/she cannot simply generate the legitimate signature σ' of block m' based on σ_1 and σ_2.

3.4. Merkle Tree. Merkle hash tree (MHT) is an authentication structure built based on hashes of data. The leaf node of Merkle tree stores the hashes of data elements (a file or a collection of files). The nonleaf node stores the hashes of its child nodes. MHT can identify whether the data were altered by comparing the calculated root hash with the value held by the validator. In blockchain network, MHT is used to store transaction's hash and check transaction's authenticity.

Figure 2 shows block structure in blockchain. Each block header saves the root hash of all transaction t_i in this block. The root hash participates in the hash operation of block header, and thus any modification to transaction data will lead to the change of the root hash, which will result in the hash change of the block header. In this paper, the user's operational information and metadata information of files are put into the blockchain. The chain structure of the block ensures the security of auditing data source.

3.5. Security Assumptions. The security of our new public auditing scheme will be based on the CDH assumption and DL assumption.

3.5.1. Computational Diffie–Hellman (CDH) Problem. Let G be a multiplicative cyclic group. The order of G is a large prime p. The generator of G is g. The CDH problem is described as follows: given two random numbers $a, b \in Z_p$ and $(g, g^a, g^b) \in G$, compute the value $g^{ab} \in G$.

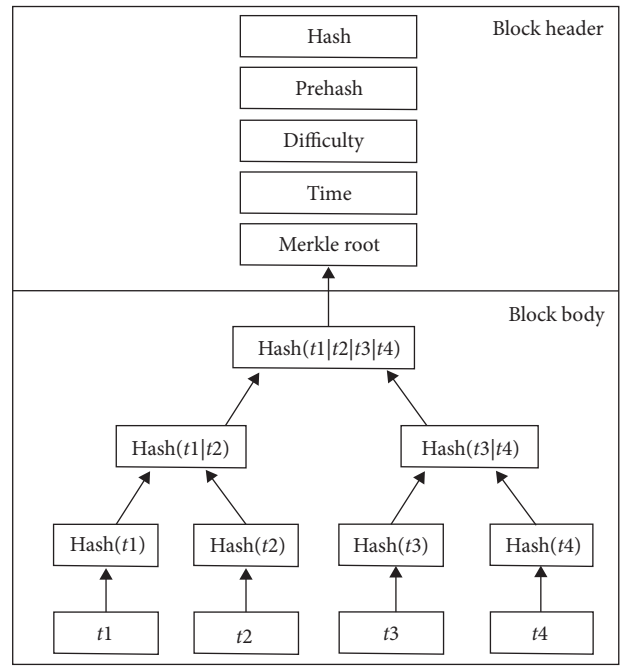

Figure 2: Block structure in blockchain.

Definition 1. **CDH assumption:** the probability that any probabilistic polynomial-time adversary $\mathcal{A}$ solves the CDH problem can be negligible, namely,

$$\Pr\left(A_{\mathrm{CDH}}\left(g, g^a, g^b \in G\right) \longrightarrow g^{ab} \in G: \forall a, b \underset{R}{\in} Z_p\right) \leq \varepsilon. \tag{1}$$

In other words, it is computationally feasible to solve the CDH problem or impossible to solve the CDH problem in a limited time.

3.5.2. Discrete Logarithm (DL) Problem. Let G be a multiplicative cyclic group. The order of G is a large prime p. The generator of G is g. The DL problem is described as follows: given $h \in G$, compute $a \in G$, such that $h = g^a$.

Definition 2. **DL assumption:** the probability that any probabilistic polynomial-time adversary $\mathcal{A}$ solves the DL problem can be negligible, namely,

$$\Pr\left(A_{\mathrm{DL}}(g, h \in G) \longrightarrow a \in Z_p, s.t.\ h = g^a\right) \leq \varepsilon. \tag{2}$$

In other words, it is computationally feasible to solve the DL problem or impossible to solve the DL problem in a limited time.

4. Public Auditing Scheme Based on SCPK

Then, we describe how to construct our public auditing scheme based on the SCPK system in more detail.

(1) *System Initialization Phase.* Let G_1 and G_2 be two groups of a large prime order p and g be a generator of

G_1. Let e be a bilinear map with e: $G_1 \times G_1 \longrightarrow G_2$. Let h be a hash function expressed as h: $\{0,1\}^* \longrightarrow G_1$. Suppose that the outsourced file F is divided into n data blocks, i.e., $F = \{m_1, m_2, \ldots, m_n\}$. Assume the identities of the user and file are ID_1 and ID_2, respectively.

(2) *Key Generation Phase*. In the system, TPA can be used as a trusted authority who is responsible for the user's registration and the generation of user's public key. TPA first publishes the modulus N and its public key pk. The private key of TPA is sk. The length of N is more than 1024 bits, and $\mathrm{pk} \times \mathrm{sk} = \varphi(N)$, where $\varphi(\cdot)$ is Euler's totient function. Then, the user selects a random number $a \in Z_p$ as his private key and calculates $v = g^a \mathrm{mod} N$. After that, the user sends the v and his identity ID_1 to TPA who will calculate user's public key $y = (v - \mathrm{ID}_1)^{h(\mathrm{ID}_1)^{-1}} \bmod N$ and send y to user. After receiving y, the user verifies the validity of equation $v = (y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1) \mathrm{mod}\ N$. If the equation holds, then the running result of this stage is $\{\mathrm{SK}, \mathrm{PK}\} = \{(a), (,)\}$, where u is the random element of G_1.

(3) *Signature Generation Phase*. With the public parameter g and his private key a, the user generates a signature $\sigma_i = h(v_i t_i) \cdot (g^{h(m_i)})^a$ for each data block m_i. The mentioned v_i is m_i's version number, and t_i is m_i's time stamp. Then, let the signature set of all blocks be $\sigma = \{\sigma_i, i \in [1,n]\}$.

(4) *File Tag Generation Phase*. To ensure the integrity of the unique file identifier ID_2, the user computes the file tag $\vartheta = TQ\mathrm{ID}_1\mathrm{ID}_2\mathrm{SIG}\{\mathrm{sk}, TQ\mathrm{ID}_1\mathrm{ID}_2\}$ with his private key $\mathrm{sk} = a$. In the equation, $T = g^x \mathrm{mod}\ N$ and $Q = g^{x-a} \mathrm{mod}\ N$, where $x \in Z_p$ is a random number chosen by the user.

Finally, the user sends the data information $\mathrm{ID}_2, \{v_i, t_i, i \in [1,n]\}$ to the TPA for auditing and uploads $F, \sigma, \vartheta, \mathrm{ID}_1$ to the CSP for storage.

(5) *Block Tag Generation Phase*. After receiving F, σ, ϑ, CSP further generates a tag $\theta_i = e(\sigma_i, g)$ for each block m_i by using the bilinear map e. Then, CSP stores the verification metadata $\vartheta, \theta = \{\theta_i, i \in [1,n]\}$ along with the file $F = \{m_1, m_2, \ldots, m_n\}$.

(6) *File Identifier Check Phase*. The user delegates the verification task of a certain file to the TPA. Then, TPA requests the corresponding file tag ϑ from CSP and verifies the equation $Q(y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1) \mathrm{mod}\ N = T$ with user's public key y. If the verification fails, TPA informs the user that the files have been corrupted; otherwise, verification continues.

(7) *Challenge Generation Phase*. TPA launches the verification challenge to the CSP in this stage. TPA first chooses a random number $k \in Z_p$ and calculates $K = g^k$, which is called random masking and is used to achieve privacy preserving [39]. Then, TPA sends the challenge information $\mathrm{chal} = \{\mathrm{idx}_i, r_i, K, i \in [1,c]\}$ to CSP, where idx_i is the index of the blocks to be checked, $r_i \in Z_P$ is the random number, and c is the selected number of the blocks to be checked [12].

(8) *Proof Generation Phase*. After receiving the challenge information, CSP would generate corresponding proofs of required blocks, which contain two parts: the tag proof and the data proof. More specifically, CSP generates the tag proof as follows:

$$T = \prod_{i \in [1,c]} \theta_i^{r_i}, \tag{3}$$

which can indicate the tags' correctness. And CSP generates the data proof as follows:

$$D = e(t, K)^M, \tag{4}$$

where $M = \sum_{i \in [1,c]} r_i \cdot h(m_i)$ and $t = y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1$. The data proof can indicate the data's integrity. Then, CSP sends the proof $\{T, D\}$ to TPA.

(9) *Proof Verification Phase*. After receiving the proof, TPA would check whether the proof is valid. More concretely, TPA checks whether

$$D \cdot e\left(\prod_{i \in [1,c]} h(v_i | t_i)^{r_i}, K\right) \stackrel{?}{=} T^k, \tag{5}$$

holds. If the above verification equation holds, it shows that the outsourced data in the cloud are integral; otherwise, it shows that the data are incomplete.

The correctness of the above equation can be demonstrated as follows:

$$
\begin{aligned}
D \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, K\right) &= e\left(t, g^k\right)^M \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, K\right) \\
&= e\left(t^M, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) \\
&= e\left(g^{a\cdot M}, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) \\
&= e\left(g^{a\cdot \sum_{i\in[1,c]} r_i \cdot h(m_i)}, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i|t_i)^{r_i}, g^k\right) \\
&= e\left(\prod_{i\in[1,c]} g^{h(m_i)\cdot a\cdot r_i}, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) \\
&= e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i} \cdot g^{h(m_i)\cdot a\cdot r_i}, g^k\right) \\
&= \prod_{i\in[1,c]} e\left(h(v_i t_i) \cdot g^{h(m_i)\cdot a}, g\right)^{r_i k} \\
&= \prod_{i\in[1,c]} \theta_i^{r_i k} \\
&= T^k.
\end{aligned}
\tag{6}
$$

5. Security Proof and Performance Analysis

In the proposed public auditing scheme, CSP is assumed to be an untrustworthy party and TPA is considered credible but curious. CSP may conceal the data errors or deliberately delete some data. TPA may be curious about the privacy information of users' data and even may try to derive the users' data contents. Then, necessary security and performance analyses of the new scheme will be comprehensively demonstrated in this section.

First of all, let us analyze the security of the self-certified public key system.

If an attacker attempts to retrieve the user's secret key a from his/her public key y, he/she must calculate the secret key from the equation $g^a = (y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1) \bmod N$. In this way, he/she will face the difficulty of computing discrete logarithm modulo N. In other words, the attacker's probability of success is to solve the discrete logarithm problem and factorization problem. Moreover, even TPA knows v and ID_1; the difficulty for him to retrieve the user's secret key a is also equivalent to the difficulty of computing discrete logarithm.

Another scenario is that an attacker tries to derive user's secret key a from the user's signature. For the file tag $\vartheta = TQ\mathrm{ID}_1\mathrm{ID}_2\mathrm{SIG}\{\mathrm{sk}, TQ\mathrm{ID}_1\mathrm{ID}_2\}$, the attacker should obtain x from $T = g^x \bmod N$ or $Q = g^{x-a} \bmod N$. However, the difficulty for him to achieve it is also equivalent to the difficulty of computing discrete logarithm problem. For the block signature $\sigma_i = h(v_i t_i) \cdot (g^{h(m_i)})^a$, the attacker should compute a from the equation. He also faces the difficulty of computing discrete logarithm problem.

The final scenario is that an attacker tries to impersonate the signer to forge a valid signature without knowing the signer's secret key a. For the file tag, the above analysis shows that the attacker cannot reveal the user's secret key. Then, he cannot forge a valid signature that can pass the verification. For the block signature, Definition 2 indicates that the probability that any probabilistic polynomial-time adversary $\mathcal{A}$ solves the DL problem can be negligible. Then, it is computationally infeasible for the attacker to forge a valid HVA in a limited time. The proof, which is demonstrated in the security analysis of [12], is omitted in this paper.

Secondly, we discuss the unforgeability of proofs.

In the presented public auditing scheme, CSP sends the proof $\{T, D\}$ to TPA after the proof generation phase. The above analysis shows that the tag proof T cannot be forged owing to the CDH assumption. Then, we only need to prove that the data proof cannot be forged. Suppose CSP sends a fake proof $\{T, D^*\}$ to TPA, where $D = e(t, K)^{M^*}$ and $M^* = \sum_{i\in[1,c]} r_i \cdot h(m_i^*)$. If CSP wants to pass the verification, the equation

$$e\left(g^{a\cdot\sum_{i\in[1,c]} r_i\cdot h(m_i)}\cdot\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) = e\left(g^{a\cdot\sum_{i\in[1,c]} r_i\cdot h(m_i^*)}\cdot\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right), \tag{7}$$

must hold. Then, we can deduce that $\sum_{i\in[1,c]} r_i \cdot h(m_i^*) = \sum_{i\in[1,c]} r_i \cdot h(m_i)$ according to the properties of bilinear maps. However, this contradicts the above assumption. That is to say, the data proof is unforgeable. In summary, our presented scheme can effectively resist against the forging attacks launched by CSP.

Thirdly, we discuss the communication and computation overhead, which are reduced by introducing the batch auditing.

With the batch auditing, multiple verification tasks from different users can be handled concurrently. Suppose that TPA sends d challenges to CSP. Then, the tag proof T_j and the data proof D_j are calculated separately. And CSP figures out the aggregate proofs according to the following equation:

$$\begin{aligned} T_B &= \prod_{j=1}^{d} T_j, \\ D_B &= \prod_{j=1}^{d} D_j, \end{aligned} \tag{8}$$

where $T_j = \prod_{i\in[1,c]} \theta_{ji}^{r_{ji}}$, $D_j = e(t_j, K_j)^{M_j}$, $t_j = y^{h(\mathrm{ID}_j)} + \mathrm{ID}_j$, and $M_j = \sum_{i\in[1,c]} r_{ji} \cdot h(m_{ji})$. ID_j is the identity of the j-th user. Then, CSP sends the aggregate proofs $\{T_B, D_B\}$ to TPA. Once received, TPA checks whether the equation

$$D_B \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}||t_{ji}\right)^{r_{ji}}, K_j\right) \stackrel{?}{=} T_B^{k_j}, \tag{9}$$

holds. v_{ji} and t_{ji} are the version number and time stamp of block m_i for the j-th user. k_j is the random number chosen by TPA for the j-th user. $K_j = g^{k_j}$ is the random masking calculated by TPA for the j-th user. $r_{ji} \in Z_P$ is the random number chosen by TPA for the j-th user.

If the above verification equation holds, it shows that our scheme can realize the batch auditing. Then, its correctness can be demonstrated as follows:

$$\begin{aligned} D_B \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, K_j\right) &= \prod_{j=1}^{d} e\left(t_j, K_j\right)^{M_j} \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, K_j\right) \\ &= \prod_{j=1}^{d} e\left(t_j^{M_j}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(g^{a_j\cdot M_j}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(g^{a_j\cdot\sum_{i\in[1,c]} r_{ji}\cdot h(m_{ji})}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} g^{h(m_{ji})\cdot a_j\cdot r_{ji}}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}} \cdot g^{h(m_{ji})\cdot a_j\cdot r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} \prod_{i\in[1,c]} e\left(h\left(v_{ij}t_{ji}\right) \cdot g^{h(m_{ji})\cdot a_j}, g\right)^{r_{ji}k_j} \\ &= \prod_{j=1}^{d} \prod_{i\in[1,c]} \theta_{ji}^{r_{ji}k_j} \\ &= \prod_{j=1}^{d} T_j^{k_j} \\ &= T_B^{k_j}. \end{aligned} \tag{10}$$

Finally, our new scheme is based on the self-certified public key system. Compared with other public auditing schemes [10, 12, 18, 19, 21–28, 38, 39], there is no public key certificate included in the public authentication parameters. And there is no need to store and transmit the public key certificate before the interaction of auditing. Then, the validation and validity of public key certificate is omitted. The verification of public key is hidden in the process of the verification of signature. Consequently, the storage space and communication bandwidth are saved. The network load and transmission delay are reduced. The verification efficiency of public and the authentication efficiency of the scheme are improved.

6. Discussion and Conclusions

In this paper, we present a public auditing protocol with a self-certified public key system using blockchain technology, which differs from the state-of-the-art schemes. The user's operational information and metadata information of the file are formed to a block after verified by the checked nodes and then to be put into the blockchain. The chain structure of the block ensures the security of auditing data source. Comprehensive analyses show that attackers cannot derive user's secret key in the proposed scheme. TPA cannot derive users' data from the collected auditing information during the verification phase. Attackers cannot impersonate the signer to forge a valid signature without knowing the signer's secret key. The presented scheme can also effectively resist against the forging attacks launched by CSP. The realization of batch auditing and the efficiency of the scheme are also discussed in this paper. Compared with other public auditing schemes, the storage space and communication bandwidth are saved in our public auditing scheme. The network load is also reduced. In addition, the verification efficiency of public key and the authentication efficiency of the scheme are improved.

However, in the actual cloud storage environment, a lot of various data need to be updated dynamically motivated by various application requirements. For instance, users might try to perform insertion operation owing to the incomplete outsourced data or might try to delete some data that are no longer used. Our public auditing scheme does not specifically discuss dynamic data auditing, which can be referred to DHT [19] or put forward as a new structure in our future research. Furthermore, TPA may dishonestly perform public auditing protocols and may even collude with CS to deceive users. Some existing public audit schemes use blockchain to resist against malicious TPA. However, CS may guess the challenge messages, and there is a risk that user information may be disclosed to TPA during the audit process. The above questions will be the focus of our future research.

Data Availability

All data, models, and codes generated or used during the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant no. 61702316, the Natural Science Foundation of Shanxi Province under grant nos. 201801D221177 and 201901D111280, the Educational Research Projects of Young and Middle-Aged Teachers in Fujian Education Department under grant no. JAT170142, the Key Research and Development Project of Shandong Province under grant no. 2019JZZY010134, and the Graduate Education Reform Research Project of Shanxi Province under grant no. 2020YJJG145.

References

[1] H. Wang, Z. Zheng, L. Wu et al., "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[2] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, "Cloud computing: opportunities and challenges," *IEEE Potentials*, vol. 33, no. 1, pp. 34–36, 2014.

[3] K. Yang and X. Jia, "Data storage auditing service in cloud computing: challenges, methods and opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[4] Z. Xia, X. Wang, X. Sun et al., "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[5] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[6] J. Shen, T. Zhou, X. Chen et al., "Anonymous and traceable group data sharing in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[7] J. Ryoo, S. Rizvi, W. Aiken et al., "Cloud security auditing: challenges and emerging approaches," *IEEE Security & Privacy*, vol. 12, no. 6, pp. 68–74, 2014.

[8] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 12, no. 2, pp. 2039–2042, 2018.

[9] Z. Fu, X. Wu, C. Guan et al., "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[10] C. Wang, S. S. M. Chow, Q. Wang et al., "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[11] C. Wang, K. Ren, W. Lou et al., "Toward publicly auditable secure cloud data storage services," *IEEE Network*, vol. 24, no. 4, pp. 19–24, 2010.

[12] Q. Wang, C. Wang, K. Ren et al., "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.

[13] J. Li, Y. K. Li, X. Chen et al., "A hybrid cloud approach for secure authorized deduplication," *IEEE Transactions on*

Parallel and Distributed Systems, vol. 26, no. 5, pp. 1206–1216, 2015.

[14] M. Mowbray, "The fog over the grimpen mire: cloud computing and the law," *Scripted*, vol. 6, p. 132, 2009.

[15] A. Juels and B. S. Kaliski Jr, "PORs: proofs of retrievability for large files," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 584–597, Acm, New Delhi, India, March 2007.

[16] 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, Acm, New Delhi, India, March 2007.

[17] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2013.

[18] Y. Zhu, G. J. Ahn, H. Hu et al., "Dynamic audit services for outsourced storages in clouds," *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227–238, 2013.

[19] 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.

[20] 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.

[21] J. Shen, J. Shen, X. Chen et al., "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.

[22] C. Wang, Q. Wang, K. Ren et al., "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceeding of the Infocom, 2010 proceedings IEEE*, pp. 1–9, San Diego, CA, USA, March 2010.

[23] Y. Zhu, H. Hu, G. J. Ahn et al., "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.

[24] C. C. Erway, A. Küpçü, C. Papamanthou et al., "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 4, p. 15, 2015.

[25] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste et al., "Efficient remote data possession checking in critical information infrastructures," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1034–1038, 2008.

[26] H. Shacham and B. Waters, *Compact Proofs of retrievability*, pp. 90–107, Springer, Berlin, Germany, 2008.

[27] W. Chen, H. Tian, C. C. Chang et al., "Adjacency-hash-table based public auditing for data integrity in mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

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

[29] R. Ding, Y. Xu, J. Cui et al., "A public auditing protocol for cloud storage system with intrusion-resilience," *IEEE Systems Journal*, vol. 2019, no. 99, pp. 1–12, 2019.

[30] N. Garg, S. Bawa, and N. Kumar, "An efficient data integrity auditing protocol for cloud computing," *Future Generation Computer Systems*, vol. 109, pp. 306–316, 2020.

[31] C. Li, J. Hu, K. Zhou, Y. Wang, and H. Deng, "Using blockchain for data auditing in cloud storage," in *Cloud Computing and Security. ICCCS 2018. Lecture Notes in Computer Science*, X. Sun, Z. Pan, and E. Bertino, Eds., Springer, Berlin, Germany.

[32] L. Linn and M. Koo, "Blockchain for health data and its potential use in health it and health care related research," *Journal of Medical Internet Research*, vol. 2016, 2016.

[33] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "Npp: a new privacyaware public auditing scheme for cloud data sharing with group users," *IEEE Transactions on Big Data*, vol. 99, 2017.

[34] S. Ghoshal and G. Paul, "Exploiting block-chain data structure for auditorless auditing on cloud data," in *ICISS 2016*, I. Ray, M. S. Gaur, M. Conti, D. Sanghi, and V. Kamakoti, Eds., pp. 359–371, Springer, Berlin, Germany, 2016.

[35] Y. Fu, "Meta-key: a secure data-sharing protocol under blockchain-based decentralised storage architecture," 2017, http://arxiv.org/abs/1710.07898.

[36] Y. Miao, Q. Huang, M. Xiao et al., "Decentralized and privacy-preserving public auditing for cloud storage based on blockchain," *IEEE Access*, vol. 8, p. 1, 2020.

[37] S. Li, J. Liu, G. Yang et al., "A blockchain-based public auditing scheme for cloud storage environment without trusted auditors," *Wireless Communications and Mobile Computing*, vol. 2020, no. 9, pp. 1–13, 2020.

[38] M. Girault, *Self-certified Public keys*, pp. 490–497, Springer, Berlin, Germany, 1991.

[39] C. Liu, R. Ranjan, X. Zhang et al., "Public auditing for big data storage in cloud computing--A survey," in *Computational science and engineering (CSE), 2013 IEEE 16th international conference on. IEEE*, pp. 1128–1135, Sydney, Australia, December 2013.

[40] B. Wang, B. Li, H. Li et al., "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *Journal of Systems and Software*, vol. 113, pp. 130–139, 2016.

[41] Q. Lin, H. Yan, Z. Huang et al., "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, 2018.

[42] Panda, "Public auditing for shared data with efficient user revocation in the cloud," *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 92–106, 2015.