A Public Auditing Framework Against Malicious Auditors for Cloud Storage Based on Blockchain

Song Li(✉), Jian Liu, and Guannan Yang

College of Information Engineering,
Nanjing University of Finance and Economics, Nanjing, China
lisong@nufe.edu.cn

Abstract. In the cloud storage applications, the cloud service provider (*CSP*) may delete or damage the user's data. In order to avoid the responsibility, *CSP* will not actively inform the users after the data damage, which brings the loss to the user. Therefore, increasing research focuses on the public auditing technology recently. However, most of the current auditing schemes rely on the trusted third public auditor (*TPA*). Although the *TPA* brings the advantages of fairness and efficiency, it cannot get rid of the possibility of malicious auditors, because there is no fully trusted third party in the real world. As an emerging technology, blockchain technology can effectively solve the trust problem among multiple individuals, which is suitable to solve the security bottleneck in the *TPA* based public auditing scheme. This paper proposed a public auditing scheme with the blockchain technology to resist the malicious auditors. In addition, through the experimental analysis, we demonstrate that our scheme is feasible and efficient.

Keywords: Cloud storage · Pubic auditing · Blockchain

1 Introduction

With the rapid development of the cloud computing, users can access the cloud services more economically and conveniently today: for example, the cloud users can outsource the numerous computing tasks to the *CSP* and reduce the purchase of local hardware resources [1]; besides, with the help of cloud storage service such as, Amazon, icloud, Dropbox, etc. [2], users can put aside the geographical restrictions and upload the local data to the *CSP*, with only a small amount of payment but a greatly reduction of local storage resources and more convenience of the data sharing with others. For the enterprise users, due to the explosive growth of business data, enterprises need to spend high cost to purchase software/hardware resources to build an IT system and maintain a professional technical team to manage this system, which causes extra burden to enterprises. Hence, the "pay as you go" service mode of the cloud storage is more convenient and practical. Users can dynamically apply for the storage space according to their data volume from the *CSP*, so as to avoid resource waste through the elastic resource allocation mechanism.

Although the cloud storage service has a broad market prospect, there are still many data security problems to be solved. Many famous *CSP* have experienced information

D. Wang et al. (Eds.): SPNCE 2020, LNICST 344, pp. 286–300, 2021.
https://doi.org/10.1007/978-3-030-66922-5_20

disclosure and service termination [3], such as icloud's information disclosure, Amazon cloud's storage outage, Intuit's power failure, sidekick's cloud disaster, Gmail's email deletion, etc. In August 6, 2018, Tencent cloud admitted to the user's silent error caused by the firmware version of the physical hard disk, i.e. the data written is inconsistent with the data read, which damages the system metadata [4]; therefore, solving the data integrity problem not only can enhance the user's confidence in the cloud storage services, but also can effectively promote the development of the cloud storage services industry. Since the cloud computing has become the basic infrastructure at the era of big data, the data security is the primary concern of cloud users.

However, in the practical applications, due to system vulnerabilities, hacker attacks, hardware damage, human operation errors or even to maximize the interests, *CSP* may delete or damage some user's data [5–7]. For example, the hospital outsourced all the electrical disease records to the *CSP*, but *CSP* may lose part of the stored data. It will cause a great loss to the users when these records cannot be retrieved. In order to avoid responsibility, the *CSP* may not actively inform the data owners after the data is damaged; in addition, in some special service models, *CSP* claims to provide multi-backup storage service, but in the actual process, they only provide ordinary single-backup storage service, and cheat the consumers to obtain additional service fees. All of these factors will cause the cloud users unable to trust the *CSP* fully.

The traditional method of checking the integrity of remotely stored files is to download all the data from the *CSP* to the local machine, then data owner checks it locally by computing the message authentication code or signature [8–11]. However, if the large amount of data has been stored in the remote cloud server, such as for the online retailer

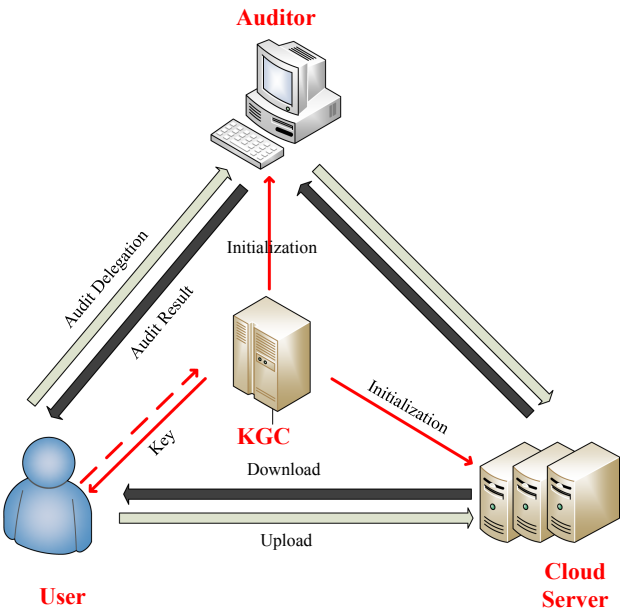

Fig. 1. System model of the public audit scheme based on trusted third party

like Amazon that produced the hundreds of *PB* data every day, it is unrealistic to download all these data to the local machines every time when checking the integrity, because this will cause a lot of bandwidth/storage resources waste; on the other hand, the integrity checking is a periodicity task, it is expensive for mobile devices with limited resources to execute locally [12]; for the fairness at last, it is not reasonable to let either part of the *CSP* or data owners to audit after the data corruption, so it is an ideal choice to introduce a trusted third party to replace *CSP* or data owners to check the data integrity [13] (Fig. 1). However, after the third-party auditor (*TPA*) has been introduced, the problem of privacy disclosure is also produced. For example, the malicious auditor obtains the data owner's identity information in the auditing process, so as to know which part of the stored data is more valuable to the user [14]; in addition, it is possible for the *TPA* to know the content of the stored data block in the interaction with *CSP* [15].

2 Related Works

In 2003, Deswarte et al. [8] proposed a remote data integrity checking scheme based on the challenge-response protocol for the distributed system. Although their scheme does not need to download all the data when checking the remotely stored data, their scheme causes a large number of modular exponential operations on the server side resulting in large computing overhead, besides, the client needs to maintain all the data backup locally. In 2004, Sebe et al. [9] proposed a remote integrity checking scheme based on the Diffel-Hellman protocol. In their scheme, the client needs to store n-bits data for each data block to be stored, that is to say, only when the size of the data block is much larger than n, their scheme has practical significance (otherwise, it is not better than storing all the data locally); in 2005, Opera et al. [10] proposed a scheme based on the tweakable encryption. However, the client needs to download all the files in the checking phase, and their scheme aims at data retrieval, which is not suitable for the scenario of data integrity checking. In 2006, Schwarz et al. [11] solved the data security problem of remote storage across multiple servers based on algebraic signature. However, the security cost in the client side increases dramatically with the increasing of the data blocks to be checked.

The proposed schemes introduced above have the same problem: the client needs to access the complete data back-up, however, it is not suitable in practice obviously as mentioned before. Many scholars have carried out research on this issue later. In 2007, Ateniese et al. [16] proposed the concept of provable data possession (PDP) firstly based on RSA homomorphic linear authenticator and random sampling technology. User can check the data stored in the remote server without downloading all the data to the local thus solve the defect existed in the early proposed schemes, however, their scheme only supports of the static data. In 2008, Shacham and Water proposed two improved schemes based on BLS short signature [17]: the first scheme based on BLS signature supports infinite times public verifications on the data; the second scheme calculates the authenticators using pseudo-random function but does not support public verification.

Except of the static data, users may also add, delete, or modify the remote data, these dynamic operations will change the index of the data block resulting in the invalid of the original authenticators, as shown in Fig. 2. If all the authenticators will be recalculated each time when the data owner performs dynamic operations, a lot of computing and

communication cost will be produced. Therefore, many scholars studied on the dynamic data supported schemes. In 2008, Ateniese et al. [18] proposed the dynamic PDP scheme based on symmetric key firstly. However, for the reason that their scheme is based on symmetric encryption, it does not support public auditing. In reference [19], Erway et al. introduced a dynamic PDP scheme that can support dynamic data using rank-based skip lists technology. In reference [20], Zhu et al. proposed a scheme with indexing-hash table to support the effective update of the dynamic data.

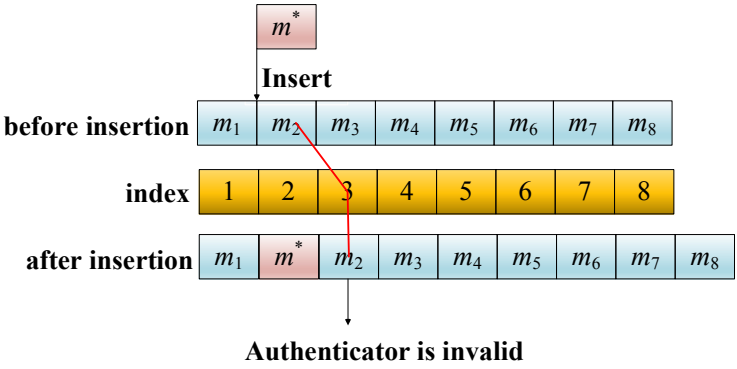

Fig. 2. The invalid of authenticator caused by the data dynamic operation (insertion)

In 2011, Hao et al. [21] expanded the scheme of Sebe et al.'s scheme [9], and proposed a dynamic auditing scheme in block-level based on RSA homomorphic tag. The so-called block level dynamic means that the data owners can insert, delete or update data blocks, but after the update, they still need to recalculate the authenticators which is not flexible.

In the practical applications, the integrity checking task is performed by *TPA* and most of the schemes proposed later support public auditing. In 2009, Wang et al. [13] proposed the integrity checking scheme with *TPA* firstly based on BLS short signature and MHT (Merkel hash tree). In this scheme, any entities in the network can challenge the *CSP* to check the integrity of the data stored on the cloud server, but this scheme does not support the full dynamic operations on the data.

Although the introduction of the *TPA* brings many benefits, it also brings new security and privacy issues. Therefore, the public auditing scheme supporting privacy preserving has become a hotspot recent years. In 2010, Wang et al. [14] proposed a public auditing scheme supporting content privacy preserving based on random mask technology. This scheme supports batch verification of multi-user tasks. However, due to the large number of verification tags generated on the server side, the system suffers a large storage burden. In 2012, Wang et al. [15] proposed a public auditing scheme to protect the identity privacy of the group users based on group signature technology, but the group signature produced huge computing cost in the data owner side, and their scheme did not consider the situation that the users can leave and join the group dynamically. In their scheme, users need to recalculate the authenticators of all the stored data block when the group key has changed; in 2014, Wang et al. [22] proposed an auditing scheme based on ring

signature technology, which can protect the identity privacy of group membership and support group members to join/leave the group dynamically, but the efficiency of their scheme will be decreased with the increasing number of the group members, and the malicious users cannot be tracked in their scheme.

In the process of authenticator generation phase, a large number of signature operations are involved, however, many of the existing terminal equipment are embedded devices with low-power capacity such as mobile phones or sensors in IoT applications, therefore, public auditing schemes for low-power equipment have also been studied: in 2015, He et al. [23] proposed a public auditing scheme based on the certificateless cryptosystem, and applied it into the cloud-assisted wireless body area networks. Based on their certificateless mechanism, certificates do not need to be transferred and stored comparing with the previous proposals thus reduced the bandwidth resources; the users does not need to do the CRL(certificate revocation list) querying which greatly saves the computing resources. In 2016, Li et al. [12] proposed two auditing schemes for low-performance equipments based on online-offline signature technology. In the first basic scheme, the *TPA* needs to store some offline signature information, so it is only suitable for users to upload some short data (such as phone number, etc.) in the cloud; in the second scheme, the author solved the problem that the *TPA* needs to store a large number of offline signatures.

In 2017, Li et al. [24] pointed out that most of the existing schemes are based on the PKI infrastructure and the security of these schemes depend on the security of the key, then Li et al. proposed a public auditing scheme based on fuzzy identity signature technology. In this scheme, the user's identity (ID) is the public key, which improves the security of the system. However, Xue et al. [25] pointed out that Li's scheme can't resist malicious auditor's attack; Yu et al. put forwarded a scheme to resist key disclosure attack in the literature [26], which guarantees the forward security of the system by supporting the key updating mechanism, and the updated keys can still audit the previous data block tagged with the old keys.

In 2013, Liu et al. [27] proposed a public auditing scheme based on the rank-based Melkel-hash tree to improve the efficiency of the traditional hash tree algorithm. However, this algorithm causes a lot of computation cost to the *TPA*. If there are a large number of data blocks, the *TPA* needs to spend a lot of time to calculate the path of the Melkel tree. Yang et al. [28] proposed a scheme based on index table structure and BLS Signature algorithm, which supports the PDP mechanism of full dynamic data operation. In their scheme, because the index table is used to store the metadata of block file through continuous storage space, the deletion and insertion move a large number of data. With the expansion of user data scale and the increase of the number of block files, the time cost of deletion and insertion will increasing dramatically, which directly leads to the increasing of verification time cost after dynamic operation and reduces the auditing efficiency. In 2016, Li et al. [29] proposed a PDP auditing model based on LBT structure (large branching tree proofs of data possession, LPDP) to solve the problem of the authentication path is too long in building the MHT. LBT adopts a multi-branch path structure, and the depth of the LBT to be constructed decreases with the increasing of out-degree, thus reducing the auxiliary information in the process of data integrity checking, simplifying the process of data dynamic update, and reducing the calculation

overhead between entities in the system. In 2017, Garg et al. [30] added indexes and timestamps to the MHT structure introduced in the scheme of reference [13] and proposed rist-MHT (relative indexed and time staged Merkle hash tree) structure, based on this structure, they proposed a PDP mode. Compared with the MHT structure, rist-MHT structure shortens the authentication length in MHT, thus reduces the time cost of node query. On the other hand, time stamp attribute gives the authenticator data freshness. However, although these algorithms based on MHT hash tree [13, 27, 30] avoid downloading all the data in the auditing process, but the correct verification results can only prove that the cloud server stores the hash tree but not the uploaded data.

In recent years, many scholars have carried out researches on the other issues such as group user revocation, data de-duplication, sensitive information sharing and anti-quantum attack etc.

In 2018, Zhang et al. [31] pointed out that in the existed group sharing schemes, user revocation results in the large computational cost of the authenticator associated with the revoked users, so they proposed an identity-based public auditing scheme that can support user revocation, in which the revoking of malicious user does not affect the auditing of the previous data blocks.

Taek-Young Youn etc. [32] combined the ciphertext de-duplication technology with public auditing scheme. Because a large number of data uploading work are transferred to the *CSP*, the client only needs to carry out a single tag calculation step, which is suitable for low-performance client environment.

Shen et al. [33] proposed a public auditing scheme that can hide sensitive information when data owner sharing the data with other users based on IBE (identity-based-encryption). In this scheme, a role of data transfer (sanitizer) is added to transfer the sensitive data and its signature to realize the privacy preserving of the sensitive information in shared medical record.

In 2019, Tian et al. [34] pointed out that up to now, none of the schemes above can meet all the security properties and put forward a new scheme. In the process of tagging, the user's signatures will be converted into group signature, thus protecting the identity privacy of the users; in the auditing process, the content privacy is protected by using mask technology; all data operations will be recorded in the operation history table so that all illegal activities can be tracked.

Xue et al. [25] proposed a public auditing scheme based on blockchain to resist malicious auditors. In their scheme, the challenge verification information is generated based on bit-coin algorithm. However, the final auditing result of their scheme still relies on *TPA* uploading to the blockchain, which does not eliminate the threat of malicious *TPA* fundamentally.

Through the analysis above, we can see that the proposed schemes have the following defect presently: the security of these schemes relies on the trusted third party - *TPA*. Although the *TPA* brings advantages of the fairness and efficiency to the auditing process, it cannot get rid of the possibility of the malicious auditor, because there is no completely trusted third party in the real world. Although some scholars have conducted research on privacy protection problem in *TPA* based public auditing schemes with group signature, ring signature and other privacy protection technologies, the *TPA* needs to be treated as a semi-trusted entity and the risk of malicious auditor have not be eliminated

fundamentally. As a new technology, blockchain technology can effectively solve the trust problem among multiple individuals, which is suitable to solve the security bottleneck problem in the *TPA* based public auditing scheme. This paper intends to solve the malicious auditor problem in the public auditing schemes combined with blockchain technology. The main contributions are summarized as follows:

1) We proposed a framework of public auditing scheme without trusted third part based on blockchain and given a basic work-flow;
2) We proposed a certificateless public auditing scheme based on the proposed framework to resist the malicious auditor and key escrow problems;
3) We gave a proven security analysis on our proposed schemes, the efficiency and security properties comparison shows that our scheme is better than previous schemes.

3 Preliminaries

In this section, we introduce the cryptographic techniques used to construct our scheme.

Definition 1: Bilinear Map

Given a cyclic multiplicative group G with order q and another multiplicative cyclic group G_T with the same order q. A bilinear pairing refers to a map e: $G \times G \rightarrow G_T$ should satisfy the following properties:

1) **Bilinearity**: For all $P, Q \in_R G$ and $a, b \in_R \mathbf{Z}_q^*$, $e(a \cdot P, b \cdot Q) = e(P, Q)^{ab}$.
2) **Non-degeneracy**: There exist $P, Q \in_R G$ such that $e(a \cdot P, b \cdot Q) \neq 1_{GT}$.
3) **Computability**: For all $P, Q \in_R G$, there exists an efficient algorithm to compute $e(a \cdot P, b \cdot Q)$.

Definition 2: Elliptic Curve Discrete Logarithm Problem (ECDLP)

Suppose that P and Q are two points over elliptic curve $E_p(a, b)$, and we know that P and Q has the relationship of $Q = s \cdot P$, it is difficult to find out the integer $s \in \mathbf{Z}_q^*$ only with P and Q.

Definition 3: Computational Diffel Hellman (CDHP)

Suppose that P, $a \cdot P$ and $b \cdot P$ are three points over elliptic curve $E_p(a, b)$, it is difficult to compute the result $a \cdot b \cdot P$ only with P, $a \cdot P$ and $b \cdot P$.

4 The Framework of Public Auditing Scheme Based on Blockchain

4.1 System Model

In our proposed framework, there are four roles: cloud server provider (*CSP*), client, key generating center (*KGC*) and auditors.

(1) *Cloud Service Provider*: In our scheme, the *CSP* is a semi-trusted entity with strong computing/storage resources, and the client uploads the local data to the remote CSP for storage. The CSP will faithfully complete the whole process of our auditing protocol with the other entities, however, he/she will attempt to cover up the fact of data corruption.
(2) *Client*: The client is a cloud storage service user. He/she could store his/her data in the *CSP* to reduce the storage burden locally. To ensure the integrity of the remotely stored data, the client can delegate the auditor to execute the interactive protocol with *CSP* and get the auditing result from auditor.
(3) *KGC*: The *KGC* is a trusted entity in our proposal, which is mainly used to generate the public parameters of the whole system and the client's partial secret key in certificateless cryptosystem.
(4) *Auditor*: auditors are distributed nodes deployed on the blockchain nodes, the *ProofVerify* algorithm are deployed on the auditors as the form of smart contract. After get the proof generated by the *CSP*, the auditors calculate the checking result and store them into the storage layer of blockchain.

The relationship among these entities is shown in Fig. 3.

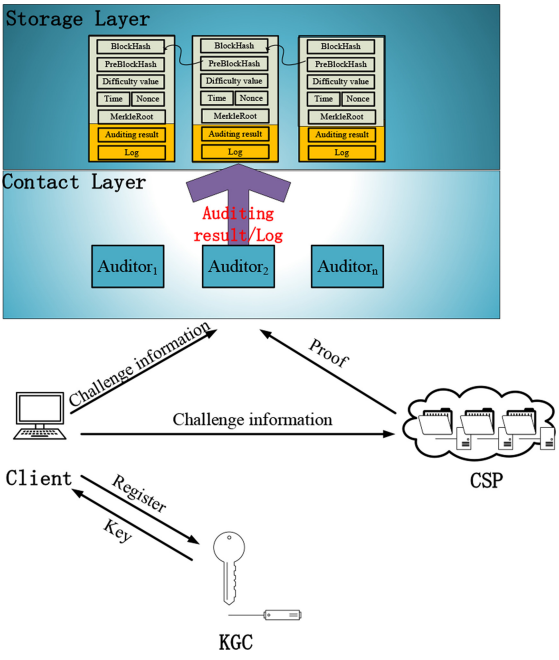

Fig. 3. The proposed framework against malicious auditors for cloud storage based on the blockchain

4.2 The Proposed Framework

In this section, we proposed a basic framework of public auditing scheme based on blockchain technology, and give a general work flow. In our framework, in order to avoid the problem of malicious attackers in the traditional *TPA* based schemes, we use the distributed nodes in the blockchain network as auditors to check the integrity.

Before the client uploads the data to the *CSP*, it uses the private key issued by the *KGC* to calculate the linear authenticator of the file. The calculation process divides the file into data blocks for calculation firstly, and then the user uploads the data and the corresponding linear authenticator to the *CSP* for storage. When the clients wants to check the integrity of the stored data in the cloud, the client sends the challenge information (randomly generated integers) and sends it to the auditors and *CSP*; the *CSP* calculates the proof according to the challenge information and returns proof to the auditors.

Auditors are smart contracts deployed on the blockchain nodes, the function of which mainly includes of two parts: processing client auditing request and execute the *ProofVerify* algorithm (the main part of the auditing scheme). The distributed auditors calculates the auditing results according to the proof returned by the *CSP*, stores the results into the storage layer of the blockchain, and maintain a history that cannot be tampered.

Secondly, when the client performs the data updating operations (such as adding, deleting, querying and modifying) on the stored data, the *CSP* generates the client's operation log of this time and compute the multiple–signatures on this log by client and *CSP* which indicate that all members agree with this result. It should be noted that auditing is a periodic process, it can be arranged every day at a certain fixed period of every day such as after zero clock, but each time the user performs an updating operation, an auditing action will be triggered automatically.

If the client or *CSP* finds out the stored data has been damaged, they can compare the current auditing results with the previous historical records stored in the blockchain, and combine the signed operation logs to determine the responsibility for data damage; because these data are stored in the distributed ledger with non-repudiation and non-tampering, neither party can refuse to admit it.

5 The Detailed Scheme

In this section, we give a detail proposal based on the framework we introduced above. Our scheme is constructed based on He et al.'s CLPA [24] scheme and Xue et al.'s scheme IDBA [26].

Setup: Input the security parameter κ, The *KGC* generates the system parameters and the master key executes the following steps:

1) The *KGC* selects a large prime number q, an additive group G_1, and uses the bilinear group generator to generate the bilinear group G_2. The *KGC* chooses a bilinear pairing e: $G_1 \times G_1 \rightarrow G_2$.

2) Let P be a generator of group G_1. The *KGC* selects a big integer $s \in \mathbf{Z}_q^*$ randomly as the master key, keeps s secretly, computes the public key $P_{pub} = s \cdot P$
3) The *KGC* publishes the system parameters $\boldsymbol{Para} = \{q, G_1, G_2, P, \mathrm{e}, h_1(\cdot), h_2(\cdot), h_3(\cdot), H_1(\cdot), H_2(\cdot), P_{pub}\}$.

PartialPrivateKeyExtract: The client registers with the *KGC* to extract the partial private key with the following steps:

1) Client submits his/her identity ID_U to the *KGC*.
2) After received the client's identity ID_U, the *KGC* chooses a random big integer $t_U \in \mathbf{Z}_q^*$ and computes $T_U = t_U \cdot P$, $h_U = h_1(ID_U, T_U)$, and $s_U = t_U + \mathrm{s} \cdot h_U \bmod q$.
3) The *KGC* sends the partial private key $D_U = \{s_U, T_U\}$ to the user secretly.

SetSecretValue: The client sets his/her secret value as follows.

1) The client chooses a big integer x_U randomly as his/her secret value.
2) The client keeps x_U secretly.

SetPublicKey: The client sets his/her public key as follows.

1) The clients computes $P_U = x_U \cdot P$.
2) The clients sets $pk_{\mathrm{U}} = \{T_{\mathrm{U}}, P_{\mathrm{U}}\}$ as his/her public key.

SetPrivateKey: The client sets $ssk_U = \{s_U, x_U\}$ as his/her private key.

Store: The client O with identity ID_O, private key $sk_O = \{s_O, x_O\}$, and public key $pk_O = \{T_O, P_O\}$ runs this algorithm to generate the tags for the data file F. Firstly, the data file F should be divided into n blocks $\{m_1, m_2, \ldots, m_n\}$, for every data blocks m_i, the client compute the tags with the following steps, where $i \in \{1, 2, \ldots, n\}$.

1) The client computes $k_O = h_2(ID_O, pk_O, P_{pub})$ and $Q = H_1(P_{pub})$.
2) The client computes $S_i = (s_O + k_O \cdot x_O)(r \cdot H_2(m_i) + H_2(id_i) + m_i \cdot Q)$ and sends $\{m_i, id_i, S_i, R\}$ to the *CSP*, where id_i is the unique identity of m_i, r is a random number,

$$R = r \cdot (T_O + h_O \cdot P_{pub} + k_O \cdot P_O) \tag{1}$$

Audit: To check the integrity of the uploaded data, the user executes the follows challenge-response protocol with *CSP*:

1) *Challen*:

The client generates challenge information as follows:

- Selects a random l-element subset $J = \{a_1, a_2, \ldots, a_l\}$ of the set $[1, n]$
- Selects a random $v_j \in \mathbf{Z}_q^*$ for each $j \in J$.

- Generate the challenge information: $Chall = \{j, v_j\}_{j \in J}$, and broadcast it in the network, *CSP* and all the auditors can get it.

2) *ProofGen*:

After receiving the challenge information $Chall = \{j, v_j\}_{j \in J}$ from the client, the *CSP* generates a proof which prove of the correctly possession of selected blocks as follows:

- Choose a big integer $x \in \mathbf{Z}_q^*$ randomly
- Computes:

$$u = x^{-1}(\sum\nolimits_{j=a_1}^{a_l} m_j \cdot v_j + h3(\sigma)) \tag{2}$$

$$\sigma = x \cdot Q \in G_1 \tag{3}$$

$$\delta = \sum\nolimits_{j=a_1}^{a_l} v_j \cdot S_j \tag{4}$$

- Broadcast the proof information $Prof = \{\delta, u, \sigma, R\}$ to the auditors; if the auditing client choose to use in the more efficient model, the *CSP* divides the data blocks into k parts, and send them to the k auditors separately, k means the number of auditors.

ProofVerify: Upon receiving the $Prof = \{\delta, u, \sigma, R\}$, the auditors execute this algorithm to check the integrity of the data stored in the *CSP*. Here, the *Prof* indicates the proof generated by *CSP*; in the secure model, the *Prof* is the proof information of all the data blocks; in the efficient model, *Prof* is the partial proof information, we use the same express as *Prof* here.

1) The auditors computes $h_O = h_1(ID_O, T_O)$, $k_O = h_2(ID_O, pk_O, P_{pub})$, and $Q = H_1(P_{pub})$.
2) The auditors checks whether equation:

$$\mathrm{e}(\delta, P) = \mathrm{e}(\sum_{j=a_1}^{a_1} v_j \cdot H_2(id_j), T_O + h_O \cdot P_{pub} + k_O \cdot P_O)$$

$$\cdot \mathrm{e}(\sum_{j=a_1}^{a_1} v_j \cdot H_2(m_j), R) \cdot \mathrm{e}(u\sigma - h_3(\sigma)Q, T_O + h_O \cdot P_{pub} + k_O \cdot P_O) \tag{5}$$

holds, and output *1* if the equation holds that represents the correct storage of the data File F; otherwise, output *0* to indicate data corruption.
3) Create an **entry**(t, *nonce*, *Chall*, *Prof*, 0/1) and broadcast it in the network, all the auditors get the full auditing result and store the result; in the secure model, for the reason that each auditor can calculating the auditing result by themselves, the broadcast operation is not needed.

DataUpdate: when the user updates the file in the cloud, a recording log *Log* is generated by *CSP* to record the details of the user's operation. *CSP* and user execute the *MultiSign*(*Log*) and broadcast it in the blockchain network for storage. After each data update operation, the system automatically triggers the *Audit* phase.

6 Security Requirements Discussions

This section discussed that our proposed scheme satisfies the security requirements of auditing schemes.

1) *Publicly verifiability*: through the correctness proof part, we can see that as long as the client correctly calculates the data tags before uploading the data file, the auditor can perform interactive algorithm with the *CSP*, and get the real storage situation of the data blocks without the help of the client. Therefore, we say that our scheme has achieved the property of publicly verifiability.
2) *Privacy preserving*: We can see that in the process of the data auditing, the auditors can only get the aggregated data blocks and the tags, but through these information, auditors can not get any available information about stored data. Therefore, we say that our scheme achieves the goal of privacy protection.
3) *Batch auditing*: through the derivation of the correctness analysis, we can see that in the process of the auditing phase, multiple data blocks can be sampled at one time, and multiple data auditing tasks can be batch verified to improve the auditing efficiency. Therefore, we say that our scheme achieves the goal of the batch auditing.
4) *Key escrow resistant:* similar to the scheme IDBA [26], our scheme is based on the certificateless cryptography, the secret key to generate the authenticator has two parts which is derived from the KGC and client respectively. Therefore, the KGC cannot get the full of user's secret key like the scheme CLPA [24] based on the identity cryptosystem.
5) *Malicious auditor resistant:* in our auditing scheme, the auditing result is calculated by the distributed nodes, none of them can tamper the auditing result only if the attacker controls 51% nodes in the network; compare to the existing blockchain based public auditing scheme [26], the *ProofVerify* phase is transferred to the blockchain as the form of smart contract, instead of relying on the third-party auditor to upload the auditing result to the blockchain, thus, the possibility of the auditor creates the false result is eliminated fundamentally; besides, for the reason that the data blocks are confused with the mask code and the auditors can get nothing about the auditing data, the privacy of the data content has been protected.

7 Experimental Analysis

This section compares the performance of our proposed scheme with that of He et al.'s CLPA [24] scheme and the scheme IDBA [26]. Table 1 is the notation list we used in Table 2.

Table 1. The notations for operations

Symbol	The time cost of corresponding operation
T_M	The point multiplication operation in G_1
T_p	The pairing operation
T_H	Hash to point function
T_h	Hash function
k	The number of auditors

Table 2. The computation cost comparison of our scheme with CLPA and IDBA

Scheme	User's computational cost	Auditing computational cost	Communication cost
CLPA [24]	$2nT_M + (n+1)T_H + T_h$	$2T_p + (n+3)\,T_M + (n+1)T_H + 2T_h$	$\|\mathbf{Z_q}\| + \|\mathbf{G_1}\|$
IDBA [26]	$3nT_M + nT_H + nT_h$	$3T_p + (2n+3)T_M + nT_H + (n+1)T_h$	$\|\mathbf{Z_q}\| + 3\|\mathbf{G_1}\|$
Ours	$(3n+3)T_M + (2n+1)T_H + T_h$	$4T_p + ((2n+4)T_M + 2nT_H + T_h)/k$	$\|\mathbf{Z_q}\| + 3\|\mathbf{G_1}\|$

Table 2 shows the security overhead of these schemes in *Store* phase on the client side and the *ProofVerify* phase on the auditors' side. From the Table 2, we can see that in the *Store* phase, the time consumption of the authenticator calculation in our scheme is slightly higher than the other two schemes.

In the *ProofVerify* stage, because we used the distributed auditors to audit the data blocks, we get the better efficiency than the other schemes. We can see that if we do not use distributed auditors for auditing tasks, the computing cost of our scheme is still the highest, but after using the distributed processing mechanism in the efficient model, the efficient has been improved greatly.

Communication Cost: In the three schemes, the challenge information is the same; in the response phase, the proof returned by our scenario is: $Prof = \{\delta, u, \sigma, R\} = |\mathbf{Z_q}| + 3|\mathbf{G_1}|$. Through the comparison of Table 2, we can find that our scheme has the same communication cost with IBDA and slightly higher than CLPA.

8 Conclusion

In this paper, we pointed out that most of the *TPA* based public auditing schemes cannot resist the malicious auditor. To solve this problem, we proposed a public auditing framework with blockchain technology and certificateless crptography. In this framework, we used the distributed nodes in the blockchain network as auditors to check the integrity

and the checking results will be stored into the storage layer of the blockchain with the tamper-resistant manner; the client operations on the data will be recorded as log signed by the data owners and *CSP* which indicate that all members agree with this result. Anyone can check the historical records stored in the blockchain nodes, and combine with the signed operation logs to determine the responsibility for data damage. We gave a detailed proven security proof of our scheme. A comprehensive performance evaluation shows that our scheme is more feasible and efficient than similar schemes.

References

1. Armbrust, M., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
2. Feng, D.-G., Zhang, M., Zhang, Y., et al.: Study on cloud computing security. J. Softw. **22**(1), 71–83 (2011)
3. Shen, W., Yu, J., Xia, H., et al.: Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium. J. Netw. Comput. Appl. **82**, 56–64 (2017)
4. http://www.sohu.com/a/245553016_671058
5. Ren, K., Wang, C., Wang, Q.: Security challenges for the public cloud. IEEE Internet Comput. **16**(1), 69–73 (2012)
6. Song, D., Shi, E., Fischer, I., Shankar, U.: Cloud data protection for the masses. IEEE Comput. **45**(1), 39–45 (2012)
7. Juels, A., Oprea, A.: New approaches to security and availability for cloud data. Commun. ACM **56**(2), 64–73 (2013)
8. Deswarte, Y., Quisquater, J.-J., Saïdane, A.: Remote integrity checking. In: Jajodia, S., Strous, L. (eds.) Integrity and Internal Control in Information Systems VI. IIFIP, vol. 140, pp. 1–11. Springer, Boston, MA (2004). https://doi.org/10.1007/1-4020-7901-X_1
9. Sebe, F., Martinez-Balleste, A., Deswarte, Y., et al.: Time-bounded remote file integrity checking. Technical report 04429 (2004)
10. Oprea, A., Reiter, M.K.: Space-efficient block storage integrity. In: Network and Distributed System Security Symposium, NDSS 2005, San Diego, California, USA. DBLP (2005)
11. Schwarz, T.S.J., Miller, E.L.: Store, forget, and check: using algebraic signatures to check remotely administered storage. In: IEEE International Conference on Distributed Computing Systems. IEEE (2006)
12. Li, J., Zhang, L., Liu, J.K., et al.: Privacy-preserving public auditing protocol for low-performance end devices in cloud. IEEE Trans. Inf. Forensics Secur. **11**(11), 2572–2583 (2016)
13. Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling public verifiability and data dynamics for storage security in cloud computing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 355–370. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04444-1_22
14. Wang, C., Wang, Q., Ren, K., et al.: Privacy-preserving public auditing for data storage security in cloud computing. In: 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2010, 15–19 March 2010, San Diego, CA, USA. IEEE (2010)
15. Wang, B., Li, B., Li, H.: Knox: privacy-preserving auditing for shared data with large groups in the cloud. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS 2012. LNCS, vol. 7341, pp. 507–525. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31284-7_30
16. Ateniese, G., Bums, R., Curtmola, R., et al.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 598–609. ACM (2007)

17. Shacham, H., Waters, B.: Compact proofs of retrievability. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 90–107. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_7
18. Ateniese, G., Pietro, R.D., Mancini, L.V., et al.: Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks. ACM (2008)
19. Erway, C.C., Küpçü, A., Papamanthou, C., et al.: Dynamic provable data possession. ACM Trans. Inf. Syst. Secur. (TISSEC) **17**(4), 1–29 (2015)
20. Zhu, Y., Hu, H., Ahn, G., et al.: Cooperative provable data possession for integrity verification in multicloud storage. IEEE Trans. Parallel Distrib. Syst. **23**(12), 2231–2244 (2012)
21. Hao, Z., Zhong, S., Yu, N.: A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. IEEE Trans. Knowl. Data Eng. **23**(9), 1432–1437 (2011)
22. Wang, B., Li, B., Li, H.: Oruta: privacy-preserving public auditing for shared data in the cloud. IEEE Trans. Cloud Comput. **2**(1), 43–56 (2014)
23. He, D., Zeadally, S., Wu, L.: Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Syst. J. **12**(1), 64–73 (2015)
24. Li, Y., Yu, Y., Min, G., et al.: Fuzzy identity-based data integrity auditing for reliable cloud storage systems. IEEE Trans. Dependable Secur. Comput. **16**(1), 72–83 (2017)
25. Xue, J., Xu, C., Zhao, J., Ma, J.: Identity-based public auditing for cloud storage systems against malicious auditors via blockchain. Sci. China Inf. Sci. **62**(3), 32104 (2019)
26. Yu, J., Wang, H.: Strong key-exposure resilient auditing for secure cloud storage. IEEE Trans. Inf. Forensics Secur. **12**(8), 1931–1940 (2017)
27. Liu, C., Chen, J., Yang, L.T., et al.: Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates. IEEE Trans. Parallel Distrib. Syst. **25**(9), 2234–2244 (2013)
28. Yang, K., Jia, X.: An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans. Parallel Distrib. Syst. **24**(9), 1717–1726 (2013)
29. Yong, L., Ge, Y., Linan, L., Xiaofei, Z., Kun, Y.: LBT-based cloud data integrity verification scheme. J. Tsinghua Univ. (Sci. Technol.) **56**(5), 504–510 (2016)
30. Garg, N., Bawa, S.: RITS-MHT: relative indexed and time stamped Merkle hash tree based data auditing protocol for cloud computing. J. Netw. Comput. Appl. **84**, 1–13 (2017)
31. Zhang, Y., Yu, J., Hao, R., et al.: Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. IEEE Trans. Dependable Secur. Comput. **17**, 608–619 (2018)
32. Youn, T.Y., Chang, K.Y., Rhee, K.H., et al.: Efficient client-side deduplication of encrypted data with public auditing in cloud storage. IEEE Access **6**, 26578–26587 (2018)
33. Shen, W., Qin, J., Yu, J., et al.: Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. IEEE Trans. Inf. Forensics Secur. **14**(2), 331–346 (2018)
34. Tian, H., Nan, F., Jiang, H., et al.: Public auditing for shared cloud data with efficient and secure group management. Inf. Sci. **472**, 107–125 (2019)