IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 October 2023, accepted 30 October 2023, date of publication 2 November 2023, date of current version 8 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3329558

RESEARCH ARTICLE

A Blockchain-Assisted Certificateless Public Cloud Data Integrity Auditing Scheme

JIANMING DU, GUOFANG DONG, (Member, IEEE), JUANGUI NING, ZHENGNAN XU, AND RUICHENG YANG
School of Electrical Information Engineering, Yunnan Minzu University, Kunming 650504, China

Corresponding author: Guofang Dong (041007@ymu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662089, and in part by the Yunnan Minzu University Graduate Research Innovation Fund Project under Grant 2023SKY007.

ABSTRACT The utilization of cloud storage is increasingly prevalent as the field of cloud computing continues to expand. Several cloud data auditing schemes have been proposed within the academic community to guarantee the availability and integrity of cloud data. Nevertheless, several schemes rely on public key infrastructure and identity-based encryption, introducing intricate challenges associated with certificate management and key escrow. Consequently, we present a certificateless encryption-based blockchain-assisted public cloud data integrity auditing scheme for data integrity. Furthermore, our proposed scheme incorporates blockchain technology to oversee the activities of semi-trusted third-party auditors and resolve the concerns mentioned above. To enhance the efficiency of dynamic data updating and ensure data privacy security, we introduce a new data structure that combines a novel counting bloom filter and a Multi-Merkel hash tree approach. The assumption of the discrete logarithm issue determines the system's security. In contrast, the security model of the scheme is comprehensively delineated. In the part dedicated to performance analysis, we assess the scheme's functionality and computational cost within the framework of existing literature. The experimental results provide proof of the scheme's comprehensive functionality and effectiveness.

INDEX TERMS Cloud storage, certificateless encryption, dynamic updating, integrity auditing, privacy protection.

I. INTRODUCTION

The utilization of the Internet to provide efficient and secure computing and storage services is facilitated by a concept known as "cloud computing." The platform has the potential to provide consumers with a unique computing resource and data center experience, demonstrating robust scalability and meeting diverse application requirements. There are several advantages associated with cloud storage, one of which is the convenience it offers customers to access their data from any location and at any given moment [1]. Cloud storage has garnered significant attention from individuals and organizations due to its notable adaptability, efficacy, and affordability attributes. Cloud storage stands out from traditional storage systems due to its substantial storage capacity and ability to retrieve data from several locations [2].

While cloud computing offers several benefits to consumers, its rapid development also presents considerable risks. Ensuring the integrity and privacy of data stored in cloud environments poses a significant challenge of utmost importance. The Cloud Service Provider (CSP) is a prime target for malicious actors due to its role in the cloud storage architecture, wherein it maintains a substantial volume of client data in a centralized manner, resulting in considerable financial gains [3]. Despite the existence of several cloud data auditing tools [4], [5], [6], [7], instances of cloud data leakage and manipulation continue to occur sporadically [4]. This is because when users transfer data to cloud storage, they relinquish physical custody and control of the data. Cloud service providers try to protect their reputation by hiding any data-related problems [8]. Guaranteeing the confidentiality

The associate editor coordinating the review of this manuscript and approving it for publication was Rahim Rahmani.

and integrity of data in cloud environments substantially influences the evolution of cloud computing and cloud storage technologies. Hence, it is imperative to conduct remote verification of the integrity of data stored in the cloud.

Most existing cloud data integrity auditing schemes rely on public auditing mechanisms, wherein the user delegates the auditing responsibility to a third-party auditor (TPA) to alleviate their workload. However, it is essential to note that the TPA, although considered semi-trusted, may possess a vested interest in the user's data. Consequently, it is imperative to uphold data privacy during the entirety of the data auditing procedure. In the cloud storage audit scheme, incorporating a proxy server (PS) is a potential solution to aid users in data processing tasks, hence alleviating the computational burden on the user. Users can remotely change stored data by executing various operations such as modifying, deleting, inserting, and other related actions. In order to ensure the timely updating of real-time data for field testing and enable users to access updated information from the cloud server side, it is imperative to execute the dynamic operation request of data properly. This will enable users to effectively comprehend the dynamic state of monitoring data. Yan et al. [10] introduced a protocol for remote data inspection aimed at mitigating replay attacks perpetrated by malicious cloud service providers (CSPs). However, implementing this protocol using the Public Key Infrastructure (PKI) system has challenges regarding certificate administration. Li et al. [11] introduced a remote data integrity checking technique based on identification, which addresses the intricate issue of certificate management arising from the Public Key Infrastructure (PKI). The approach employs identity-based cryptography(IBC)technology, which effectively addresses the intricate challenge of certificate administration, albeit presenting a key escrow issue.

In this work, we provide a blockchain-assisted certificateless public cloud data integrity auditing scheme to approach the abovementioned problems. Considering a comprehensive audit scheme with high efficiency and security, our contribution can be summarized as follows:

1. We used blockchain technology to assist in enforcing smart contract agreements that require the semi-trusted entity TPA to do the audit work as the user asks and upload the audit record to the blockchain for the user to see.
2. Based on the novel counting bloom filter (NCBF) and Multi-Merkle hash tree (M-MHT) approaches, we create an effective and safe data structure called NCBF-M-MHT. M-MHT stores data, assures data security and provides efficient dynamic updating of the data. In contrast, NCBF allows quick data lookups and improves audit efficiency.
3. To deal with the complex certificate management and key escrow problems, we use the certificateless encryption (CE) architecture. In order to alleviate customers' computational burden, a proxy service provider is also introduced to assist users with data signing. The proposed scheme's system model and security model are both defined. The security model incorporates privacy protection, resistance to replacing attacks, and essential audit accuracy and robustness.
4. Performance and security analyses were used to evaluate the proposed scheme's security and effectiveness. The results of the performance analysis demonstrated the applicability of the proposed approach.

II. RELATED WORKS

In recent years, cloud data auditing has drawn more and more attention. By randomly selecting multiple data blocks, Ateniese et al. [12] introduced the first open auditing technique based on RSA homomorphic tags to validate the accuracy of cloud data remotely. Yang et al. [13] proposed an efficient identity-based provable data possession protocol with compressed cloud storage. In this scheme, cloud storage auditing uses only encrypted data blocks, achieved by self-authentication. It allows the reconstruction of original data blocks from outsourced data. However, this scheme does not support dynamic updating of data. Yu et al. [14] proposed a new identity-based remote cloud data auditing protocol that utilizes key homomorphic cryptographic primitives to reduce the cost of the system and the complexity of setting up and managing a public key authentication framework. Shu et al. [15] propose a blockchain-based decentralized public auditing scheme that leverages a decentralized blockchain network to take on the responsibilities of a centralized TPA and mitigates the impact of tempting auditors and malicious blockchain miners by adopting the concept of decentralized self-organization. Tian et al. [16] This paper proposes a blockchain-based secure de-duplication and shared auditing scheme for distributed storage. The scheme employs a blockchain-based two-way shared auditing mechanism to achieve decentralized public auditing without needing a TPA. Wang [17] proposes a novel remote data integrity checking model in multi-cloud storage to eliminate the complex certificate management problem. After authorization from the client, the protocol enables private, delegated, and public verification. Li et al. [18] proposed a new remote data ownership checking protocol for checking the integrity of data shared between groups using certificateless signing techniques. In this scheme, a user's private key consists of a partial key generated by the group manager and a secret value chosen by the user himself. To ensure that the correct public key is selected during data integrity checking, each user's public key is associated with his or her unique identity. This scheme does not require certificates and eliminates the key escrow problem. Zhao et al. [19] proposed a practical blockchain-assisted conditional anonymity privacy-preserving public auditing scheme that achieves resistance to man-in-the-middle attacks, storage correctness, data privacy protection, and conditional identity anonymity. Guo et al. [20] proposed a revocable blockchain-assisted ABE with an escrow-free system that solves the key escrow problem by replacing traditional key management agencies with federated blockchains.

To support the dynamic update of data, Shen et al. [21] proposed an efficient public auditing protocol for cloud data with a new dynamic structure consisting of a doubly linked info table (DLIT) and a location array (LA) that significantly reduces computational and communication overheads. Thangavel and Varalakshmi [22] proposed a cloud storage auditing scheme based on ternary hash trees (THT), which has increased dynamic update performance compared to binary trees to allow the dynamic updating of data. Wang et al. [23] explores the problem of providing public verifiability and data dynamics for remote data integrity checking in a cloud computing environment. We improve the existing storage model proofs to achieve efficient data dynamics by manipulating the classical Merkle hash tree (MHT) to construct block tag authentication. The scheme also supports multiple auditing tasks to improve auditing efficiency. A dynamic hash table (DHT) was employed by Li et al. [24] to construct an effective certificateless verifiable data ownership mechanism that also included privacy protection. An auditing method based on the Multi-Replica Position-aware Merkle Tree (MR-PMT) was presented by Peng et al. [25]. It can efficiently audit the integrity of replica files. However, its auditing efficiency declines as the number of replica files increases. The Batch-Leaves-Authenticated Merkle Hash Tree (BLA-MHT), which has its index and can fend against replacement attacks, was suggested by Rao et al. [26] in 2020. It can conduct bulk authentication on several leaf nodes.

Organization: The remainder of the paper is organized as follows: We describe specific technological preparations in Section III. The system model and the threat model are presented in Section IV. The suggested scheme's security is examined in Section V. Section VI uses simulation experiments to assess the scheme's performance. Finally, Section VII provides a summary of the whole paper.

III. PRELIMINARIES

A. BILINEAR MAPPING

A bilinear pairing [27] can map a pair of group elements into another group element. Let G_1 and G_2 both be multiplicative cyclic groups of order large prime p and g denote the generating element of the group A. A function G is named a bilinear mapping if it has the following characteristics:

1) Bilinear: For $\forall u, v \in G_1$ and $x, y \in Z_p$, there is $e(u^x, v^y) = e(u, v)^{xy}$ holds;
2) Computable: a valid algorithm for computing $e(u, v)$ exists for $\forall u, v \in G_1$;
3) Non-degenerate: there exists g such that $e(g, g) \neq 1$ holds.

B. DIFFICULT ASSUMPTIONS

The Discrete Logarithm(DL) problem [28]: probabilistic polynomial-time algorithm Λ solving the DL problem in G_1 is defined as

$$AdvDL_\Lambda = \Pr[\Lambda(g, g^a) = a, a \rightarrow Z_p] \leq \varepsilon$$

where g and g^a are used as inputs to solve for a, the successful solution of the DL problem lies in the choice of Λ and a. The DL problem is one in which the probability of computing the DL problem in G_1 is negligible for any probabilistic polynomial-time algorithm Λ.

C. MULTI-MERKLE HASH TREE(M-MHT)

The primary function of the M-MHT authenticated binary tree structure is to carry out data integrity verification, which aims to quickly and securely demonstrate if a group of components has been damaged and updated. The root node of M-MHT is referred to as such, and the root node authentication ensures all leaf nodes' integrity. The primary means of guaranteeing data security is the M-MHT root node, which may be signed by the user and kept on the server.

D. NOVEL COUNTING BLOOM FILTER

Traditional bloom filters (BF) only allow insertion and search query operations on elements; they do not support deletion operations on elements, and once data is stored in BF, data records cannot be deleted. To solve this drawback, the counting bloom filter (CBF) replaces the array of bits in BF with an array of counters, which means that each bit position is a small counter, and it allows for insert, modify, and delete operations on CBF. However, the use of traditional CBF is not enough to meet the efficiency of data structure; this paper proposes NCBF structure on the basis of CBF structure. NCBF can be associated with stored data location in addition to supporting data dynamic operations, which can greatly improve the efficiency of data dynamic processing and verification of data lookup.

IV. METHOD

A. SYSTEM MODEL

The system model of the blockchain-assisted certificateless public cloud data integrity auditing scheme is shown in Fig.1. There are five entities in this system model: Data Owner (DO), Key Generation Centre (KGC), PS, TPA and CSP.

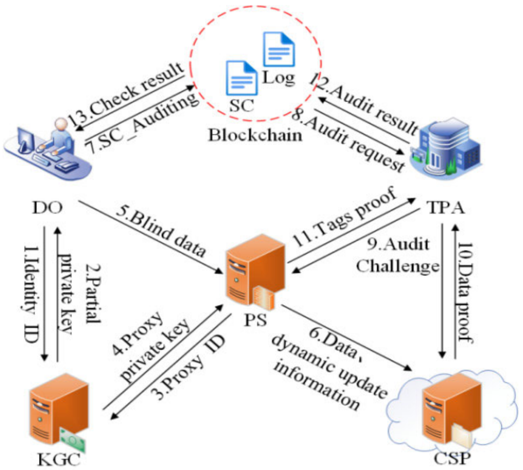

FIGURE 1. System model.

DO is the data owner who needs to upload the data to the cloud for storage but needs to blind the data information before uploading it to the proxy server to protect the data's privacy information. The proxy server helps the user sign the cloud data for uploading to the CSP for storage, which can reduce the computation overhead of DO. KGC is the key generation center that generates the partial key for the DO and the PS based on the ID of them. CSP is the not fully trusted entity that provides the DO with mighty computing power and storage space but needs to encrypt the data for storage to prevent malicious CSP from corrupting or tampering with the cloud data. CSP is not a fully trusted entity that can provide DO with computing power and storage space with solid capability. However, it must store the encrypted data to prevent malicious CSP from corrupting or tampering with the cloud data. TPA is a semi-trusted entity that can carry out the integrity auditing task on behalf of DO. However, it needs to pay attention to protect the privacy of the data during the auditing process.

B. SECURITY MODEL

The proposed scheme in this paper has the following security features: audit correctness, audit robustness, privacy protection, and resistance to substitution attacks. Security is defined as follows:

1) AUDIT CORRECTNESS

It means that only the data proof generated by CSP and the label proof generated by PS are valid simultaneously to pass the TPA verification.

2) AUDIT ROBUSTNESS

It implies that it is computationally infeasible for a CSP or PS to falsify audit certificates to pass TPA verification.

3) PRIVACY PROTECTION

It means that CSP, PS, or TPA cannot access the data content of DO in the initialization phase and audit phase.

4) RESISTANT TO REPLACE ATTACKS

CSP and PS cannot pass the TPA verification by replacing the specified data block and its signature with a substituted data block and its signature.

C. THE DETAILS OF NCBF-M-MHT

The scheme in this paper introduces M-MHT because the root node of the M-MHT structure can be signed and stored on the proxy server by the user. When a data record wants to be verified, the user does it by recalculating the signature of the M-MHT root node, ensuring the data's security. The data structure of the scheme in this paper is obtained by combining the NCBF structure and the M-MHT structure, called NCBF-M-MHT, as shown in Fig.2, which can achieve efficient dynamic data update, data insertion and deletion, as shown in Fig.3 and Fig.4.

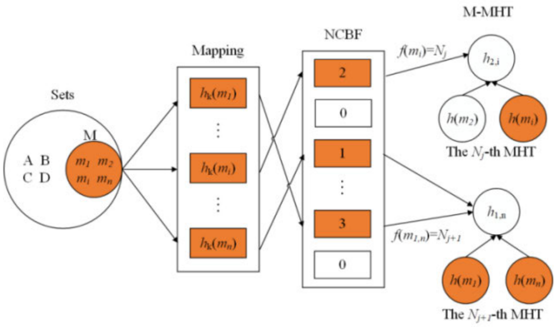

FIGURE 2. NCBF-M-MHT.

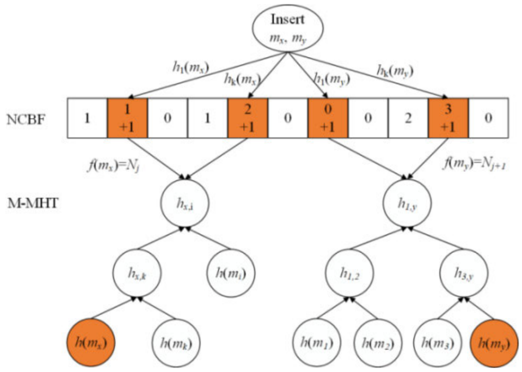

FIGURE 3. Data insertion diagram.

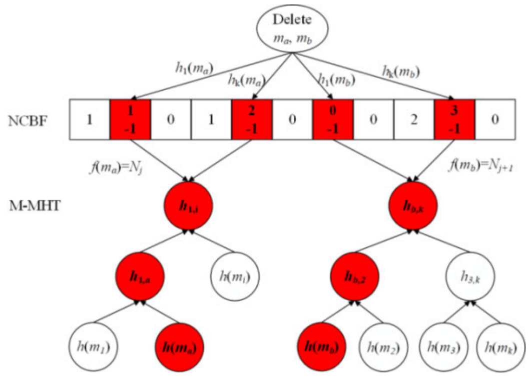

FIGURE 4. Data deletion diagram.

D. AUDIT PROTOCOL

The program consists of eight algorithms(*Setup*, *DataBlind*, *TagGen*, *DataUpload*, *ChalGen*, *ProofGen*, *ProofVerify* and *DataUpdate*). The individual algorithms are summarized as follows:

1) SETUP$(1^{\kappa}) \rightarrow$ SysPara

System initialization algorithm. Takes the system security parameter κ input and outputs the system global parameter *SysPara*.

2) DATABLIND$(M, \alpha) \rightarrow M'$

Data blinding algorithm. The plaintext data M and the blinding factor α are used as input, and the blinded data M' is output.

3) *TagGen*$(M', SysPara, u) \rightarrow \delta$

Tag generation algorithm. The blinded data M', the system parameters $SysPara$, and the proxy private key u are used as input to output the set of blinded data tags δ.

4) *DATAUPLOAD*$(M') \rightarrow T/F$

Data upload algorithm. It takes the blinded data M' as input and verifies if its data is correct; if correct, it outputs T and uploads it to the cloud; if not, it ends the storage service.

5) *ChalGen*$(M', S) \rightarrow CAHL$

Challenge generation algorithm. The blinded data M' and a subset S of challenge elements are used as input, and the audit challenge $chal$ is the output.

6) *ProofGen*$(M', \delta, CHAL) \rightarrow PROOF$

Proof generation algorithm. The blinded data M', the set of blinded data tags δ, and the audit challenge $chal$ are input to output the audit challenge proof $proof$.

7) *PROOFVERIFY*$(SysPara, CHAL, PROOF) \rightarrow TRUE/FALSE$

Proof verify algorithm. The system parameters $SysPara$, audit challenge $chal$,and audit challenge proof $proof$ are used as input, and the audit challenge results $True/False$ are output.

8) *DATAUPDATE*$(UPDATE, i, M') \rightarrow M'^{*}$

Dynamic update algorithm. The dynamic update instruction $Update$, the data block index i, and the blinded data M' are used as input, and the updated blinded data M'^{*} is the output.

E. THE DETAILS OF ALGORITHM

In this subsection, the algorithm proposed in this scheme is explained in detail. The audit process of this scheme is shown in Fig.5.

1) *SETUP*$(1^{\kappa}) \rightarrow SysPara$

KGC executes this algorithm. Two p order large prime multiplicative cyclic groups G_1 and G_2 are selected. g and β are a random generating element of the group G_1 and $g, \beta \in G_1$. The bilinear pairing function is $e : G_1 \times G_1 \rightarrow G_2$ and the secure hash function is $H : \{0,1\}^{*} \rightarrow G_1$. KGC randomly selects $\lambda \in Z_p$ as the system primary key. KGC randomly selects $u \in Z_p$ as the private key of PS according to the identity of the proxy server PS_{ID} and calculates $y = g^u$. According to the user identity DO_{ID}, KGC randomly selects $\mu \in Z_p$ as the partial private key. The final system parameter $SysPara = \{G_1, G_2, p, g, y, H\}$ is published.

2) *DATABLIND*$(M, \alpha) \rightarrow M'$

This algorithm is executed by DO. First, DO divides the data M with the file name F_{name} into n data sub-blocks, i.e., $M = \{m_1, m_2, \cdots, m_n\}$; then it calculates the blinded data block $m_i' = (m_i||i) + \alpha$, where the blinding factor $\alpha = f_{\tau}(\mu||F_{name})$ and $\tau \in Z_p$ are the key seeds of the random function f; finally, it sends the blinded data $M' = \{m_1', m_2', \cdots, m_n'\}$ to the PS.

3) *TagGen*$(M', SysPara, u) \rightarrow \delta$

This algorithm is executed by PS. PS signs the blinded data blocks

$$\delta_i = (H(m_i') \cdot \beta^{m'_i})^u \tag{1}$$

with its own private key $u \in Z_p$, the signature set $\delta = \{\delta_i\}_{1 \le i \le n}$, stores it in the dynamic data structure and finally sends $\{F_{name}, M'\}$ together to the CSP.

4) *DataUpload*$(M') \rightarrow T/F$

This algorithm is executed by CSP. Before the blinded data blocks are uploaded to the cloud storage, the data needs to be verified. CSP stores only the data blocks that pass the verification and outputs T, i.e., it calculates

$$M = \{m_i\}_{1 \le i \le n} = \{m_i'\}_{1 \le i \le n} - \alpha \tag{2}$$

and if each data block m_i corresponds to its index i, then CSP stores the blinded data block m_i'.

5) *ChalGen*$(M', S) \rightarrow CAHL$

This algorithm is executed by TPA. When DO wants to verify the data integrity in the cloud, the auditing smart contract SC_Auditing is deployed on the blockchain and the TPA performs the verification process instead of DO. First TPA selects a subset of c elements $S = \{s_1, s_2, \cdots, s_c\}$ in the set $[1, n]$, and randomly selects $v_i \in Z_p$, then the audit challenge $chal = (i, v_i)_{i \in S}$, and sends the audit challenge $chal$ to CSP and PS.

6) *ProofGen*$(M', \delta, CHAL) \rightarrow PROOF$

This algorithm is done jointly by PS and CSP. After the PS receives the audit challenge, it finds the data block to be challenged for questioning based on the index and generates the corresponding data signature proof

$$\theta = \prod_{i=s_1}^{s_c} \delta_i^{v_i} \tag{3}$$

to send to the TPA; after the CSP receives the audit challenge, it finds the data block to be challenged for questioning based on the index and generates the corresponding data proof

$$\varpi = \sum_{i=s_1}^{s_c} m_i' \cdot v_i \tag{4}$$

to send to the TPA. Then the audit proof of the audit challenge $proof = (\theta, \varpi)$.

7) *PROOFVERIFY*$(SysPara, CHAL, PROOF) \rightarrow TRUE/FALSE$

This algorithm is executed by the TPA. The TPA verifies the integrity of the cloud data based on the audit certificate $proof = (\theta, \varpi)$ and verifies if the equation

$$e(\theta, g) = \overset{?}{=} e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot \beta^{\varpi}, y) \tag{5}$$

Holds, and outputs $True$ if the equation holds and $False$ if it does not.

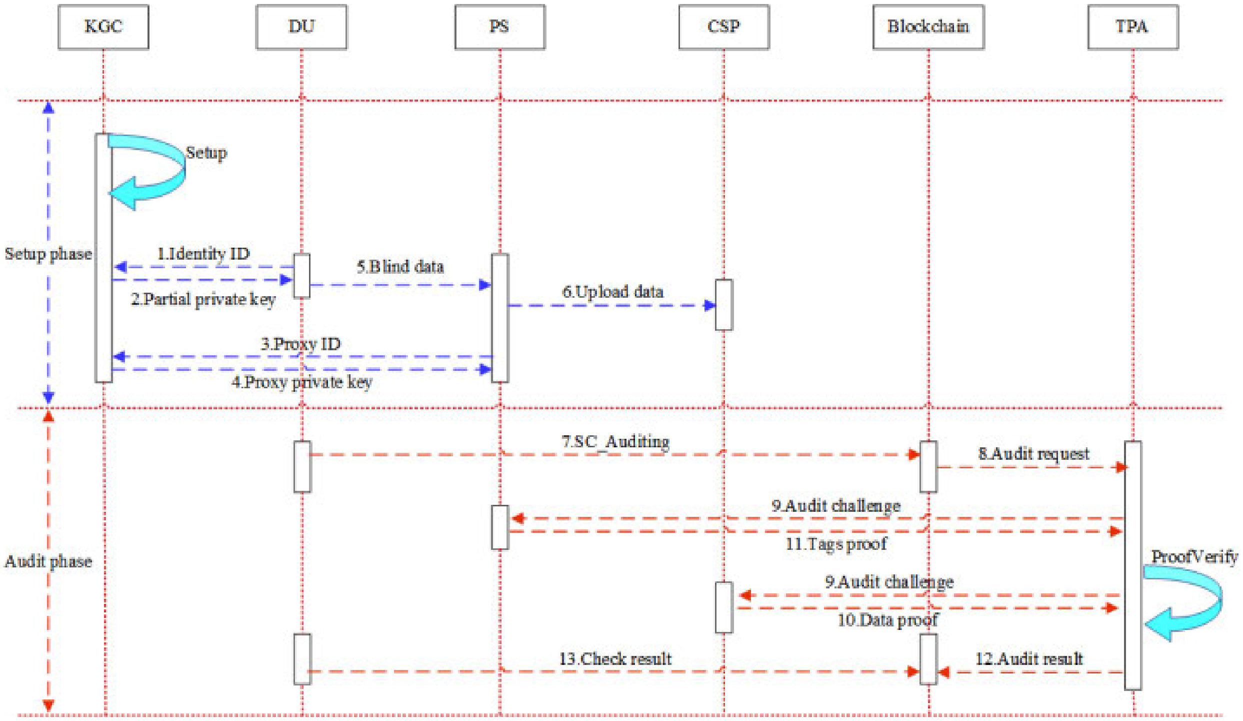

FIGURE 5. Diagram of data audit process.

8) *DATAUPDATE* (*UPDATE*, *i*, *M′*) → *M′**

The algorithm is done by multiple entities working together.

a: THE DATA MODIFICATION PROCESS IS AS FOLLOWS

Step1: When DO wants to modify the data block m_i' into a new data block $m_i'^*$, then DO generates the data modification information $Update\_M = \{Mod, i, m_i'^*\}$ and sends it to PS, where *Mod* means data modification operation and *i* is the location of the modified data block.

Step2: After receiving *Update_M*, PS locates the index number *i and* calculates the corresponding data signature $\delta_i^* = (H(m_i'^*) \cdot \beta^{m_i'^*})^u$ for the data block $m_i'^*$, then updates the count of NCBF in the data structure and modifies the node information of the corresponding MHT. Finally, $Update\_M = \{Mod, i, m_i'^*\}$ is sent to the CSP.

Step3: After the CSP receives *Update_M*, verify the validity of the data block $m_i'^*$, and store it after the verification is passed.

b: THE DATA INSERTION PROCESS IS AS FOLLOWS

Step1: If DO wants to insert a new data block $m_i^{\prime\cdot}$ after the data block m_i', then DO generates the data insertion information $Update\_I = \{Ins, i, m_i^{\prime\cdot}\}$ and sends it to PS, where *Ins* means data insertion operation and *i* is the position of the data block insertion.

Step2: After receiving *Update_I*, PS locates the index number *i* and calculates the corresponding data signature $\delta_i^{\cdot} = (H(m_i^{\prime\cdot})\cdot\beta^{m_i^{\prime\cdot}})^u$ for the data block $m_i^{\prime\cdot}$, then updates the count of NCBF in the data structure and modifies the node information of the corresponding MHT. Finally, $Update\_I = \{Ins, i, m_i^{\prime\cdot}\}$ is sent to the CSP.

Step3: After the CSP receives *Update_I*, verify the validity of the data block $m_i^{\prime\cdot}$, and store it after the verification is passed.

c: THE DATA DELETION PROCESS IS AS FOLLOWS

Step1: If DO wants to delete the data block m_i', then DO generates the data deletion message $Update\_D = \{Del, i, m_i'\}$ and sends it to PS, where *Del* indicates the data deletion operation and *i* is the location where the data block is deleted.

Step2: After receiving *Update_D*, PS locates it according to the index number *i*. Then it counts and deletes the NCBF in the data structure, and deletes the node information of the corresponding MHT. Finally, $Update\_D = \{Del, i, m_i'\}$ is sent to CSP.

Step3: After the CSP receives *Update_D*, verify the validity of the data block m_i' and delete it after the verification is passed.

V. SECURITY ANALYSIS

In this section, we evaluate the security of the proposed scheme based on audit correctness, audit robustness, data privacy protection, and resistance to replacement attacks.

Theorem 1 (Audit Correctness): Audit correctness is a fundamental requirement for cloud data auditing. Only data proof generated by CSP and label proof generated by PS are valid at the same time to pass TPA verification.

Proof: Given a valid audit certificate from CSP and PS *proof* $= (\theta, \varpi)$, the correctness of Equation (5) can be verified as follows:

$$\begin{aligned} e(\theta, g) &= e(\prod_{i=s_1}^{s_c} \delta_i^{v_i}, g) \\ &= e(\prod_{i=s_1}^{s_c} (H(m_i') \cdot \beta^{m_i'})^{u \cdot v_i}, g) \\ &= e(\prod_{i=s_1}^{s_c} ((H(m_i'))^{v_i} \cdot \beta^{\sum_{i=s_1}^{s_c} m_i' \cdot v_i}), g^u) \\ &= e\prod_{i=s_1}^{s_c} ((H(m_i'))^{v_i} \cdot \beta^{\varpi}, y) \end{aligned} \tag{6}$$

From the above equation, it can be seen that if the proof returned by CSP or PS is invalid, it will not pass the above equation verification. Therefore only label proof θ and data proof ϖ corresponding and valid at the same time can pass the TPA verification.

Theorem 2 (Audit Robustness): In this scenario, it is computationally infeasible for CSP or PS to forge audit proofs to be verified by TPA.

Proof: Define the forgery attack game as follows: Assuming the correct data block is m_i', the TPA sends a challenge query *chal* $= (i, v_i)_{i \in S}$ to the CSP and PS, and the valid audit proof returned should be *proof* $= (\theta, \varpi)$ to pass the TPA's verification. However, the CSP generates data proof $\varpi^* = \sum_{i=s_1}^{s_c} m_i'^* \cdot v_i$ for an incorrect data block $m_i'^* (m_i'^* \neq m_i')$. Define that there exists at least one $\Delta\varpi = \varpi - \varpi^*$ that is not zero on the set S. The CSP wins if the incorrect data proof still passes the verification of the TPA, and fails if the opposite is true. Assuming that the CSP wins, we have

$$e(\theta, g) = \overset{?}{=} e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot \beta^{\varpi^*}, y) \tag{7}$$

according to Equation (5). However, it is the proof *proof* $= (\theta, \varpi)$ that is the valid audit proof, so we have

$$e(\theta, g) = \overset{?}{=} e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot \beta^{\varpi}, y) \tag{8}$$

By the nature of bilinear mapping, we have $\beta^{\varpi} = \beta^{\varpi^*} \Rightarrow \beta^{\Delta\varpi} = 1$, and by the above definition, there exists at least one $\Delta\varpi$ that is not zero, so we have $\varpi \neq \varpi^*$, i.e., it is computationally infeasible for the CSP to generate the wrong data proof to pass the TPA's verification. Similarly, it follows that it is computationally infeasible for PS to generate incorrect data signature proof to pass the verification of TPA.

During the Setup phase, challenger C maintains all processed files sent to probabilistic polynomial time adversary A. After completing the last round of the audit protocol, adversary A outputs an proof that satisfies the audit challenge *chal**, which is capable of completing the validation of Equation (5) but generates at least one metadata aggregation tag that is not generated from the data maintained by challenger C.

Suppose adversary A wins the game with non-negligible probability. Construct a polynomial probabilistic time algorithm Λ, given a multiplicative cyclic group G_1 of prime order p with generating element β and a DL difficulty assumption (β, ζ), the algorithm Λ interacts with adversary A to compute χ such that it satisfies $\zeta = \beta^{\chi}$. The process is as follows:

From Equation (7) and Equation (8) we have that

$$e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot \beta^{\varpi^*}, y) = e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot \beta^{\varpi}, y)$$

a further derivation yields $\prod_{i=s_1}^{s_c} \beta^{\varpi^*} = \prod_{i=s_1}^{s_c} \beta^{\varpi}$, defined on the set S, and there exists at least one $\Delta\varpi = \varpi^* - \varpi$ that is not zero. We have

$$\beta^{\Delta\varpi} = \beta^{\sum_{i=s_1}^{s_c} m_i'^* \cdot v_i - \sum_{i=s_1}^{s_c} m_i' \cdot v_i} = \beta^{\sum_{i=s_1}^{s_c} m_i'^* \cdot v_i} \cdot \beta^{-\sum_{i=s_1}^{s_c} m_i' \cdot v_i} = 1$$

so we get the solution to the DL difficulty assumption method as follows:

$$\chi = -(\sum_{i=s_1}^{s_c} m_i'^* \cdot v_i) \cdot (\sum_{i=s_1}^{s_c} m_i' \cdot v_i)$$

where $\sum_{i=s_1}^{s_c} m_i'^* \cdot v_i \neq 0$.

Since there exists at least one $\Delta\varpi = \varpi^* - \varpi$ that is not zero, $v_i (1 \leq i \leq c)$ is a random value and with probability $\Pr[\sum_{i=s_1}^{s_c} m_i'^* \cdot v_i \neq 0] = 1/p$.

If the difference between the probability of adversary A winning the game is non-negligible, then the above algorithm Λ can be constructed to solve the DL problem.

Theorem 3 (Data Privacy Protection): During the initialization phase, the probability that the PS or CSP obtains real data information from the blinded data blocks is negligible. During the audit phase, the TPA cannot obtain the real data information from the data signature proof $\theta = \prod_{i=s_1}^{s_c} \delta_i^{v_i}$ sent by the PS and the data proof $\varpi = \sum_{i=s_1}^{s_c} m_i' \cdot v_i$ sent by the CSP.

Proof: In the initialization phase, PS receives the data block $m_i' = (m_i || i) + \alpha$ from DO after blinding, where the blinding factor $\alpha = f_{\tau}(\mu || F_{name})$ is generated based on DO's private key and randomly selected key seed, and the probability that PS wants to extract the real information of the data block is negligible. In the auditing phase, TPA receives the audit proof *proof* $= (\theta, \varpi)$ from PS and CSP, where

$$\begin{aligned} \theta &= \prod_{i=s_1}^{s_c} \delta_i^{v_i} \\ &= \prod_{i=s_1}^{s_c} (H(m_i') \cdot \beta^{m_i'})^{u \cdot v_i} \end{aligned}$$

$$= \prod_{i=s_1}^{s_c} (H(m_i')^{v_i} \cdot \beta^{\sum_{i=s_1}^{s_c} v_i \cdot m_i'})^u$$

$$= \prod_{i=s_1}^{s_c} (H(m_i')^{u \cdot v_i} \cdot (\beta^{\varpi})^u \tag{9}$$

From the above equation, $(\beta^{\varpi})^u$ is privacy-processed by $\prod_{i=s_1}^{s_c} (H(m_i'))^{u \cdot v_i}$, and the DL problem occurs during the computation of $\prod_{i=s_1}^{s_c} (H(m_i'))^{u \cdot v_i}$, while the probability of solving the DL problem in polynomial time is negligible. The only data blocks that TPA can obtain based on the data proof $\varpi = \sum_{i=s_1}^{s_c} m_i' \cdot v_i$ are also the blinded data blocks, and cannot obtain information about the real data blocks.

Theorem 4 (Resistance to Substitution Attack): In this scheme, CSP and PS cannot pass the verification of TPA by replacing the specified data block and its signature with the substituted data block and its signature.

Proof: Define the substitution attack game as follows: The TPA sends an audit challenge $chal = (i, v_i)_{i \in S}$ to the CSP and PS. They return an audit proof $proof = (\theta, \varpi)$. In the process of generating the audit proof, the CSP and PS replace the *j-th* block of information with the *k-th* block of information ($k \neq j$). The CSP and PS win if the generated audit proof still passes the TPA's verification, and fail otherwise. According to the bilinear mapping property, the left side of the Equation (5) yields that

$$\begin{aligned}
e(\theta^*, g) &= e(\prod_{i=s_1}^{s_c} \delta_i^{v_i} \cdot \delta_k^{v_k}, g) \\
&= e(\prod_{i=s_1}^{s_c} (H(m_i') \cdot \beta^{m_i'})^{u \cdot v_i} \cdot (H(m_k') \cdot \beta^{m_k'})^{u \cdot v_k}, g) \\
&= e(\prod_{i=s_1}^{s_c} (H(m_i'))^{u \cdot v_i} \cdot (H(m_k'))^{u \cdot v_k} \cdot \beta^{u \cdot (v_i \cdot m_i' + v_k \cdot m_k')}, g) \\
&= e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot (H(m_k'))^{v_k} \cdot \beta^{\sum_{i=s_1}^{s_c} (v_i \cdot m_i' + + v_k \cdot m_k')}, y)
\end{aligned}$$

Right side of the Equation(5):

$$e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot (H(m_j'))^{v_j} \cdot \beta^{\varpi^*}, y)$$

$$= e(\prod_{i=s_1}^{s_c} (H(m_i'))^{v_i} \cdot (H(m_j'))^{v_j} \cdot \beta^{\sum_{i=s_1}^{s_c} (v_i \cdot m_i' + v_j \cdot m_j')}, y)$$

Assuming that the above verification passes, we have $H(m_j') = H(m_k'), m_j' \cdot v_j = m_k' \cdot v_k$, and by the above definition $k \neq j$, then $m_j' \neq m_k'$, which means that the equations $H(m_j') = H(m_k'), m_j' \cdot v_j = m_k' \cdot v_k$ do not hold. Therefore it is computationally infeasible for CSP and PS to pass the TPA verification with the replaced data blocks.

VI. PERFORMANCE ANALYSIS

In this section, we evaluate three aspects of the proposed scheme, namely, computational overhead, communication overhead and functional comparison, from both theoretical and experimental aspects. First, we analyze the computational overhead, communication overhead, and functional comparison from the theoretical level; then we build a simulation environment for simulated experimental analysis. To further demonstrate the practicality of the proposed scheme, we compare and analyze this scheme with other cloud data auditing schemes. The definitions of the operators used are given in TABLE 1.

TABLE 1. The description of various operations.

Operations	Description
T_H	The hash mapping to G_1
T_{Exp}	The exponentiation operation on G_1
T_{Mul}	The multiplication operation on G_1
T_{Add}	The add operation on G_1
T_P	The bilinear mapping

A. THEORETICAL ANALYSIS

1) COMPUTATION OVERHEAD

The computational overhead of the proposed scheme in this paper mainly comes from the three stages of data label generation, audit proof generation and proof verification. In the data tag generation phase, the computational overhead of PS to compute the data tags $\delta_i = (H(m_i') \cdot \beta^{m_i'})^u$ is $n(T_H + T_{Mul} + 2T_{Exp})$. In the audit proof generation phase, the total computation overhead of CSP and PS to compute audit proof is $n(T_{Add} + 2T_{Mul} + T_{Exp})$. In the proof verification phase, the computation overhead of TPA to verify the audit proof is $2T_P + c(T_H + 2T_{Exp} + T_{Mul})$. Comparing this solution with other solutions in the three stages of data label generation, audit proof generation and proof verification, the results of the comparative analysis are shown in TABLE 2.

2) COMMUNICATION OVERHEAD

In this scheme, only the communication cost incurred in the audit challenge query generation phase and proof generation phase is considered. In order to meet the 160bit security of the system, the proposed scheme sets the group parameters $|G_1|$ and $|Z_p|$ to be 512bit and 160bit in size, respectively. $|p|$ and $|q|$ are the lengths of the elements on G_1 and Z_p, respectively. In the challenge generation phase of this scheme, the TPA initiates a challenge query $chal = (i, v_i)_{i \in S}$ to the CSP and PS with the communication overhead of $c(|p| + |q|)$, and the communication overhead generated by the CSP and PS returning proof $proof = (\theta, \varpi)$ to the TPA is $|p| + |q|$. TABLE 3 shows the comparison of the communica-

TABLE 2. **The computation overhead of different schemes.**

Scheme	TagGen	ProofGen	ProofVerify
Shu et al.[15]	$2n(T_H + 2T_{Mul})$	$(4n+2)T_{Mul} + T_H + T_{Add}$	$3T_P + c(2T_{Add} + 5T_{Mul} + 3T_H)$
Tian et al.[16]	$n(2T_H + T_{Mul} + 2T_{Exp})$	$n(T_{Add} + 2T_H + 2T_{Mul})$	$2T_P + 3c(T_{Exp} + T_{Mul})$
Ours scheme	$n(T_H + T_{Mul} + 2T_{Exp})$	$n(T_{Add} + 2T_{Mul} + T_{Exp})$	$2T_P + c(T_H + 2T_{Exp} + T_{Mul})$

TABLE 3. **The communication overhead of different schemes.**

Scheme	Generate challenge	Generate proof
Shu et al.[15]	$2c\|q\|$	$2(\|p\|+\|q\|)$
Tian et al.[16]	$c(2\|p\|+\|q\|)$	$2\|p\|+\|q\|$
Ours scheme	$c(\|p\|+\|q\|)$	$\|p\|+\|q\|$

tion cost between this scheme and other cloud data auditing schemes when sending the challenge set in the challenge generation phase and the audit proof in the proof generation phase.

3) FUNCTIONAL COMPARISON

In this subsection, the proposed scheme's and other schemes' functionality will be compared. The comparison results are shown in Table 4, which shows that [15] and [16] do not have dynamic update functions. Both use IBC and PKI encryption, bringing key escrow problems and complex certificate management problems. This scheme uses certificate-less encryption, which can solve the complex certificate management and key escrow problems. Compared with other schemes, this scheme introduces blockchain technology as an auxiliary means to supervise TPAs to perform cloud data integrity auditing according to DO requirements through smart contracts.

B. EXPERIMENTAL ANALYSIS

1) ON-CHAIN OVERHEAD

We tested the computational overhead of four smart contracts in a prototype Ether-based blockchain system. We evaluated our scheme based on the value of Gas consumed. On Ether, the execution of smart contracts consumes a certain amount of Gas, which is used to pay miners and guarantee the correctness of code execution. Two types of Gas are consumed during smart contract execution: Transaction-consumed Gas and Execution-consumed Gas. Transaction-consumed Gas is generated by the transaction itself and is used to pay for transactions on the blockchain network. The execution process of the contract code generates execution-consumed Gas. It is used to pay for the execution of the code.

As shown in Fig. 6, our proposed scheme has four smart contracts that must be deployed to run on the blockchain. In the check result smart contract, the value of Gas consumed is relatively small because the function is relatively simple, as it only needs to view the audit result on the blockchain. The transaction-consuming Gas and execution-consuming Gas required for the check result smart contract are 353,242 and 278,918 units, respectively. The audit smart contract sends audit challenges, verifies audit proofs, and supervises relatively complex tasks with transactions and executions consuming 835,070 and 730,566 units of Gas, respectively, which is the most Gas-consuming of the four contracts.

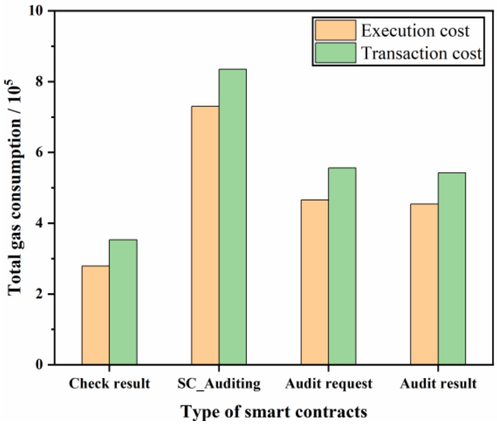

FIGURE 6. **Smart contracts on-chain overhead.**

2) OFF-CHAIN OVERHEAD

In this section, the performance of this solution will be evaluated through experiments. The experimental environment is configured with AMD Ryzen7 5800H with Radeon Graphics 3.2GHz RAM32GHz laptop, and all simulations are implemented on the Ubuntu system. The algorithms of the scheme were designed using the C programming language, and the Pairing Cryptography PBC (PBC), library version 0.5.14, and the GUN Multiple arithmetic Precision (GMP), library version 6.2.1, were used to implement the corresponding cryptographic operations. An asymmetric supersingular elliptic curve with a finite field size of 512 bits and a fixed security parameter of 160 bits is chosen.

a: TIME OVERHEAD OF THE DATA SIGNATURE GENERATION PHASE

Fig. 7 shows the time overhead performance curves of the proposed scheme with [15] and [16] in the data block signature generation phase. Compared with [15], this scheme does not have heavy multiplication operations, so its computation

TABLE 4. Functional comparison of different schemes.

Scheme	Dynamic update	Blockchain	Encryption mode
Shu et al.[15]	×	√	IBC
Tian et al.[16]	×	√	PKI
Ours scheme	√	√	CE

overhead is lower. While [16] has more Exponential operations than the present scheme, the computational overheads are higher.

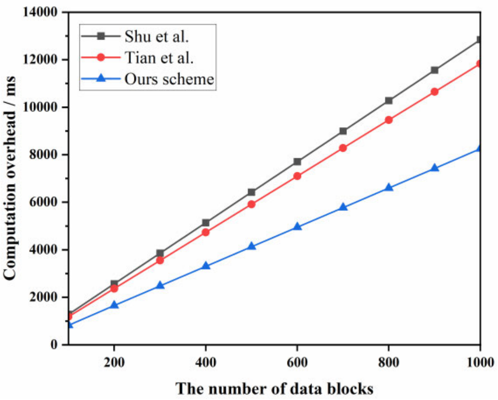

FIGURE 7. Data signature time overhead.

b: TIME OVERHEAD OF THE DATA PROOF GENERATION PHASE

The total time overhead performance curves of CSP and PS for generating corresponding data proof based on challenge interrogation are shown in Fig.7. From Fig. 8, the proof generation time for all scenarios increases linearly with the increased interrogated data blocks. Checking all data blocks in the cloud will result in more computational burden. Therefore, for efficiency reasons, specifying 460 data blocks in the challenge interrogation message applies to the actual cloud data auditing system, which can achieve at least a 99% probability of data corruption or tampering, in which case the computational overhead of this scheme only spends about 1.14s.

c: TIME OVERHEAD OF DATA PROOF VERIFICATION PHASE

The performance curve of time overhead generated by TPA during the data proof verification phase is shown in Fig. 9. From Fig. 9, all the computational overheads of the verification data are linear, increasing with the number of interrogated data blocks. However, this scheme has fewer multiplication operations, exponential operations, and pairing operations and thus uses correspondingly less verification time, which is about 8.27s for validating 1000 data blocks, compared with about 9.69s, 12.62s, and 14.92s for the other schemes.

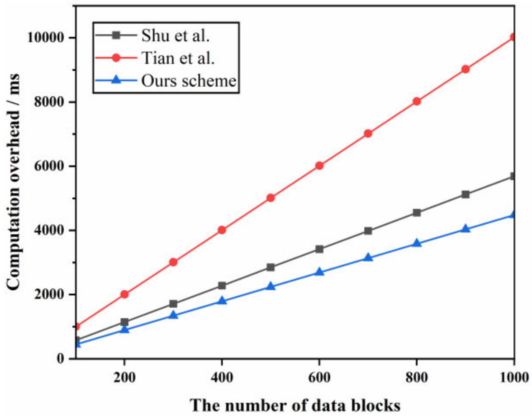

FIGURE 8. Proof generation time overhead.

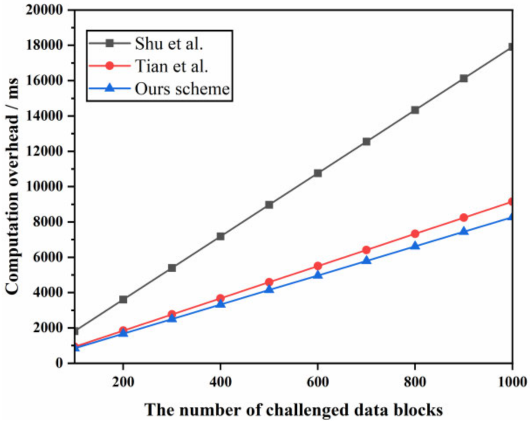

FIGURE 9. Proof verification time overhead.

d: TIME OVERHEAD OF DYNAMIC UPDATE PHASE

Fig. 10. shows the time overhead performance curves of this scheme and reference [22], [26] for the three dynamic operation phases of data modification, insertion, and deletion. The scheme in this paper uses the NCBF-M-MHT data structure with $O(1)$ time complexity for insertion and lookup of data blocks. Since the depth of the BLA-MHT structure increases exponentially with the number of data blocks, its dynamic update costs much more than this scheme. The depth of the THT structure changes more slowly as the number of data blocks increases, and its dynamic update overhead is slightly lower than that of the BLA-MHT structure. The NCBF-M-MHT structure proposed in this paper has more evident advantages in the dynamic updating of data.

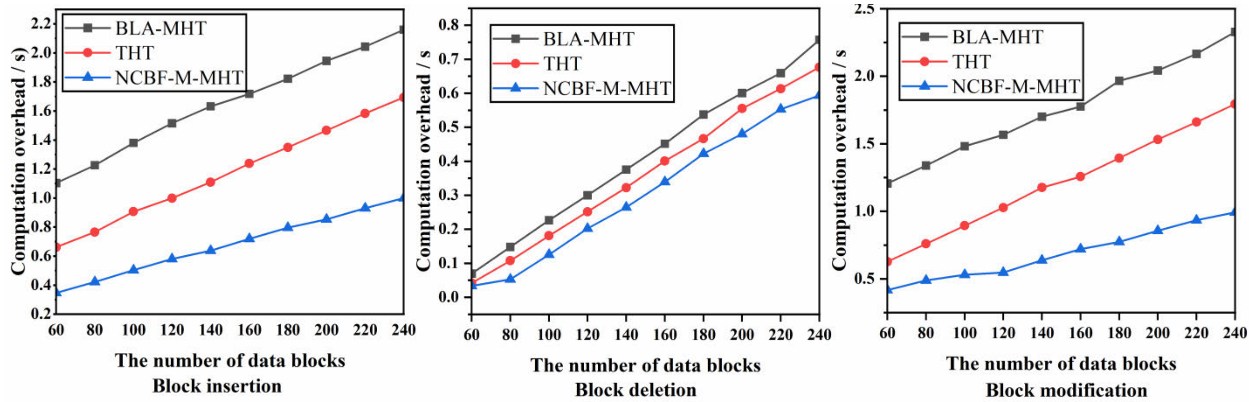

FIGURE 10. Dynamic update time overhead.

VII. CONCLUSION

This paper proposes a blockchain-assisted certificate-free public cloud data integrity auditing scheme for secure cloud storage. Our scheme uses a certificateless encryption model to eliminate the complex certificate management in PKI and key escrow in IBC. It introduces blockchain as an auxiliary means to supervise the auditing process of semi-trusted TPA and to ensure data privacy for TPA during the auditing process. A proxy server alleviates some computational overhead for users in the data initialization phase. The security of this scheme is demonstrated under DL's difficult assumptions. The performance analysis results show that this scheme is efficient and feasible.

REFERENCES

[1] H. Tian, F. Nan, C.-C. Chang, Y. Huang, J. Lu, and Y. Du, "Privacy-preserving public auditing for secure data storage in fog-to-cloud computing," *J. Netw. Comput. Appl.*, vol. 127, pp. 59–69, Feb. 2019.

[2] Y. Sun, Q. Liu, X. Chen, and X. Du, "An adaptive authenticated data structure with privacy-preserving for big data stream in cloud," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3295–3310, 2020.

[3] Y. Li, Y. Yu, B. Yang, G. Min, and H. Wu, "Privacy preserving cloud data auditing with efficient key update," *Future Gener. Comput. Syst.*, vol. 78, pp. 789–798, Jan. 2018.

[4] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[5] W. Guo, H. Zhang, S. Qin, F. Gao, Z. Jin, W. Li, and Q. Wen, "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Gener. Comput. Syst.*, vol. 95, pp. 309–322, Jun. 2019.

[6] J. Li, H. Yan, and Y. Zhang, "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 356–365, Jan. 2022.

[7] H. Yan, J. Li, J. Han, and Y. Zhang, "A novel efficient remote data possession checking protocol in cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 78–88, Jan. 2017.

[8] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1214–1226, May 2021.

[9] G. Bian, R. Zhang, and B. Shao, "Identity-based privacy preserving remote data integrity checking with a designated verifier," *IEEE Access*, vol. 10, pp. 40556–40570, 2022.

[10] H. Yan, J. Li, and Y. Zhang, "Remote data checking with a designated verifier in cloud storage," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1788–1797, Jun. 2020.

[11] J. Li, H. Yan, and Y. Zhang, "Identity-based privacy preserving remote data integrity checking for cloud storage," *IEEE Syst. J.*, vol. 15, no. 1, pp. 577–585, Mar. 2021.

[12] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.* Alexandria, VA, USA: ACM, Oct. 2007, pp. 598–609.

[13] Y. Yang, Y. Chen, F. Chen, and J. Chen, "An efficient identity-based provable data possession protocol with compressed cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1359–1371, 2022.

[14] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 767–778, Apr. 2017.

[15] J. Shu, X. Zou, X. Jia, W. Zhang, and R. Xie, "Blockchain-based decentralized public auditing for cloud storage," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2366–2380, Oct. 2022.

[16] G. Tian, Y. Hu, J. Wei, Z. Liu, X. Huang, X. Chen, and W. Susilo, "Blockchain-based secure deduplication and shared auditing in decentralized storage," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 3941–3954, Nov. 2022.

[17] H. Wang, "Identity-based distributed provable data possession in multicloud storage," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 328–340, Mar./Apr. 2015.

[18] J. Li, H. Yan, and Y. Zhang, "Certificateless public integrity checking of group shared data on cloud storage," *IEEE Trans. Services Comput.*, vol. 14, no. 1, pp. 71–81, Jan. 2021.

[19] J. Zhao, H. Huang, C. Gu, Z. Hua, and X. Zhang, "Blockchain-assisted conditional anonymity privacy-preserving public auditing scheme with reward mechanism," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4477–4488, Sep. 2022.

[20] Y. Guo, Z. Lu, H. Ge, and J. Li, "Revocable blockchain-aided attribute-based encryption with escrow-free in cloud storage," *IEEE Trans. Comput.*, vol. 72, no. 7, pp. 1901–1912, Jul. 2023.

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

[22] M. Thangavel and P. Varalakshmi, "Enabling ternary hash tree based integrity verification for secure cloud data storage," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 12, pp. 2351–2362, Dec. 2020.

[23] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847–859, May 2011.

[24] R. Li, X. A. Wang, H. Yang, K. Niu, D. Tang, and X. Yang, "Efficient certificateless public integrity auditing of cloud data with designated verifier for batch audit," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8079–8089, Nov. 2022.

[25] S. Peng, F. Zhou, J. Li, Q. Wang, and Z. Xu, "Efficient, dynamic, and identity-based remote data integrity checking for multiple replicas," *J. Netw. Comput. Appl.*, vol. 134, pp. 72–88, May 2019.

[26] L. Rao, H. Zhang, and T. Tu, "Dynamic outsourced auditing services for cloud storage based on batch-leaves-authenticated Merkle Hash Tree," *IEEE Trans. Services Comput.*, vol. 13, no. 3, pp. 451–463, May 2020.

[27] I. Kim, W. Susilo, J. Baek, and J. Kim, "Harnessing policy authenticity for hidden ciphertext policy attribute-based encryption," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1856–1870, May 2022.

[28] Y. Jiang, X. Xu, and F. Xiao, "Attribute-based encryption with blockchain protection scheme for electronic health records," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 3884–3895, Dec. 2022.

JIANMING DU is currently pursuing the M.S. degree with the School of Electrical Information Engineering, Yunnan Minzu University, Yunnan, China. His research interests include cloud computing security and privacy protection. He is a Student Member of CCF.

GUOFANG DONG (Member, IEEE) received the Ph.D. degree from the Kunming University of Science and Technology, Yunnan, China. She is currently an Associate Professor with the School of Electrical Information Engineering, Yunnan Minzu University. Her research interests include security protocols, the IoT security, and cloud computing security. She is a member of CCF.

JUANGUI NING is currently pursuing the M.S. degree with the School of Electrical Information Engineering, Yunnan Minzu University, Yunnan, China. Her research interests include information security and privacy protection.

ZHENGNAN XU is currently pursuing the M.S. degree with the School of Electrical Information Engineering, Yunnan Minzu University, Yunnan, China. Her research interests include cloud computing security and data sharing.

RUICHENG YANG is currently pursuing the M.S. degree with the School of Electrical Information Engineering, Yunnan Minzu University, Yunnan, China. His research interests include information security and cloud computing security.