Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 2330502, 11 pages
https://doi.org/10.1155/2021/2330502

Research Article

Cloud Data Integrity Verification Algorithm for Sustainable Accounting Informatization

Lin Yang [1,2]

[1]*Pricing Management Office, Cancer Hospital of China Medical University, Shenyang 110042, Liaoning, China*
[2]*Pricing Management Office, Liaoning Cancer Hospital & Institute, Shenyang 110042, Liaoning, China*

Correspondence should be addressed to Lin Yang; dengli@syu.edu.cn

Received 19 October 2021; Revised 4 November 2021; Accepted 10 November 2021; Published 30 November 2021

Academic Editor: Sang-Bing Tsai

In recent years, people have paid more and more attention to cloud data. However, because users do not have absolute control over the data stored on the cloud server, it is necessary for the cloud storage server to provide evidence that the data are completely saved to maintain their control over the data. Give users all management rights, users can independently install operating systems and applications and can choose self-service platforms and various remote management tools to manage and control the host according to personal habits. This paper mainly introduces the cloud data integrity verification algorithm of sustainable computing accounting informatization and studies the advantages and disadvantages of the existing data integrity proof mechanism and the new requirements under the cloud storage environment. In this paper, an LBT-based big data integrity proof mechanism is proposed, which introduces a multibranch path tree as the data structure used in the data integrity proof mechanism and proposes a multibranch path structure with rank and data integrity detection algorithm. In this paper, the proposed data integrity verification algorithm and two other integrity verification algorithms are used for simulation experiments. The experimental results show that the proposed scheme is about 10% better than scheme 1 and about 5% better than scheme 2 in computing time of 500 data blocks; in the change of operation data block time, the execution time of scheme 1 and scheme 2 increases with the increase of data blocks. The execution time of the proposed scheme remains unchanged, and the computational cost of the proposed scheme is also better than that of scheme 1 and scheme 2. The scheme in this paper not only can verify the integrity of cloud storage data but also has certain verification advantages, which has a certain significance in the application of big data integrity verification.

1. Introduction

In the process of enterprise development, only the realization of accounting informatization can develop enterprise informatization. Therefore, the realization of accounting informatization has become the phased goal of most enterprises. Cloud computing has experienced cross-era changes from a new product. Under the current situation, more and more enterprises have increased the development business of cloud computing, trying to combine accounting information management system with cloud computing to realize informatization [1]. Cloud computing will soon be fully applied in enterprises. The combination of cloud computing and accounting informatization means that a new system will be built on the basis of cloud technology on the network. It will be more possible for enterprises to realize informatization by using this system. But the combination of accounting informatization and cloud computing will also face some new problems. If the data are stored in the cloud, the real-time monitoring of the data will be lost. At the same time, based on the network transmission bandwidth and other reasons, users cannot frequently download the whole data to check whether the data in the cloud is preserved completely, so the integrity and security of the data are threatened [2, 3]. Message authentication refers to verifying the integrity of the message. When the receiver receives the information, it can verify that the received information has not been changed. In the traditional data verification

scheme, digital signature, digital watermark, and message authentication code are generally used to verify the integrity of data. These technologies require users to save the entire data. Therefore, if these traditional authentication methods are adopted, users need to download the whole data every time, which will bring huge communication costs to user authentication and limit the user's verification frequency [4].

In order to solve the above problems, researchers have proposed many schemes, which are generally divided into two categories: provable data possession (PDP) and proof of retrieval (POR). PDP can effectively ensure the integrity of data in cloud storage. Data integrity refers to the accuracy and reliability of data. It is proposed to prevent the existence of data that does not meet the semantic requirements in the database and prevent invalid operation or error information caused by the input and output of error information. Data integrity is divided into four categories: entity integrity, domain integrity, referential integrity, and user-defined integrity. POR can not only detect whether the stored data is complete but also recover the damaged data in the cloud by erasure code technology. Barsoum proposed a mapping-based provable multireplica dynamic data possession (MB-PMDDP) scheme. The magic transformation scheme can prove that the cloud service provider is credible by storing fewer copies, supports dynamic data outsourcing, and allows users to access the file copies stored in the cloud service provider. However, this scheme only has a limited number of queries and cannot explicitly support the operation of data block insertion [5, 6]. Data block (block) is the smallest unit for Oracle to allocate and read I/O (at least one block must be allocated, one block read, and one block written). This is a logical concept. The logical concept of the Oracle database gradually decreases from tablespace, segment, extent, and data block and can be one to many, one to many, one to many, and one by one. Omote proposes a direct repair and dynamic operation in POR based on network coding. When the server has problems, the scheme supports the direct repair of data. The user can store it in the server and use it normally, which avoids the burden of repairing data on the client. However, the scheme has strict restrictions on the size of cloud server and data storage [7]. Monarat introduces the data structure of the authentication hop table to realize the full dynamic operation of data. However, the authentication hop table needs to save too much auxiliary information, which increases the communication overhead of the overall mechanism and affects the overall performance [8].

This paper proposes an integrity public audit scheme based on the LBT authentication structure by referring to the LBT tree structure. The scheme supports the dynamic update of the single data block and the batch dynamic operation of the data block. The experimental results show that the scheme can shorten the authentication path and reduce the cost of hash operation to a certain extent.

2. Improvement of Cloud Data Integrity Verification Algorithm for Accounting Informatization

2.1. The Theoretical Basis of Cloud Computing in the Application of Accounting Informatization

2.1.1. Network Accounting Theory. Network accounting is an accounting activity that relies on the confirmation, measurement, and disclosure of various transactions and events in the Internet environment. At the same time, it is also an accounting information system based on a network environment. It is an important part of e-commerce. It can help companies realize remote processing such as financial and business collaborative remote reporting, reporting, auditing, and auditing. Network accounting is different from traditional accounting on the assumption of continuity; only for the limitations of the overall work of the enterprise, the former can more accurately analyze the authenticity of accounting information, but network accounting also has disadvantages; for example, the cooperation between network accounting and enterprises is not continuous, and network accounting analysis of the actual situation of enterprises is still lacking, which formed accounting decentralization hypothesis theory [9, 10].

2.1.2. System Theory. A system is a unified whole with special fixed goals, which is composed of two or more interacting and dependent elements. Accounting work has the characteristics of system aggregation, and as an independent system, it plays an important role in practical work. The process of fund movement is carried out under the mutual coordination and management of financial and accounting work, which can effectively combine various elements and play a certain role in the overall goal [11]. In a broad sense, liquidity refers to all the current assets of an enterprise, including cash, inventory (materials, work-in-process, and finished products), accounts receivable, securities, prepayments, and other items. The above items are all necessary for business operation, so there is a popular name for working capital, which is called operating working capital. Working capital in a narrow sense = current assets – current liabilities. According to the different needs of enterprises, accounting information can be divided into different subsystems. Each subsystem is related to each other and affects each other so as to achieve the overall goal.

2.1.3. Information Security Theory. Information security refers to protecting information resources from being damaged and making information resources relatively safe. The emergence of cloud computing technology is a protection method for information security, but it is also risk

bearing. Information security protection methods include the following: (1) Physical environment security: access control measures, regional video surveillance, fire prevention, waterproofing, lightning protection, and antistatic measures in the electronic computer room. (2) Identity authentication: two-factor identity authentication, identity authentication based on digital certificates, identity authentication based on physiological characteristics, and so on. (3) Access control: physical access control, network access control (such as network access control NAC), application access control, and data access control. (4) Audit: physical level (such as access control and video surveillance audit), network audit (such as network audit system and sniffer), application audit (implemented during application development), desktop audit (for files in the host and for system equipment), and records of operations (such as modification, deletion, and configuration). The protection method refers to the fact that the data information is uploaded to the cloud computing technology platform so that the relevant information resources are centrally stored in a database so that the data information can be protected, and the system will automatically redistribute according to the needs when using. But it is also because the virtual accounting information system composed of cloud computing technology is based on the network; that is, to use this service platform, it is necessary to transfer the data information resources to a third-party platform that cannot be seen, which will also increase the risk of information security [12]. Therefore, we should pay attention to the use of cloud computing technology to truly serve information security.

2.1.4. Cybernetics. A remarkable characteristic of modern accounting is the application of cybernetics in accounting. The unique structure of the accounting information system may cause changes in the transmission way, time, and degree of reduction of accounting information, thus affecting the quality of accounting information [13]. The characteristics of an accounting information system include the following: (1) A wide range of data sources and a large amount of data are required. (2) The structure of the data and the process of data processing are more complicated. (3) Data authenticity and high reliability are required. (4) There are many data processing links, and many processing steps are periodic. (5) Data processing has strict regulations and requires clear audit trails. (6) There are many types and large quantities of information output, and there are strict requirements on the format. (7) There are strict requirements for the security and confidentiality of the data processing process. Therefore, in the process of the target travel, more or less there will be some cases inconsistent with the original plan, which requires that the accounting work needs to use control and other means to help achieve the goal. Before the implementation of the financial plan to carry out some accounting information system prior to the control, it can be in the enterprise financial operations activities before finding out the problem, solve the problem, and timely correct deviation. In-event control refers to the control of normal economic activities in an enterprise to solve problems found in the process, so it is also known as real-time control. And postcontrol refers to the feedback of accounting, through the collection of phased accounting work information, the real feedback of accounting work. Postcontrol takes the temperature, pressure, flow, liquid level, composition, and other process parameters as the automatic control of the controlled variables. Real-time control is one way, and the main thing is to correct the deviation.

2.1.5. Cloud Accounting. The definition of China Cloud Computing Service Network refers to cloud computing products that can be used as services, including cloud host, cloud space, cloud development, cloud testing, and comprehensive products. Cloud accounting is a virtual accounting information system, which provides accounting, accounting management, and accounting decision-making services to enterprises through Internet service platform. Cloud computing is the process of decomposing huge data computing processing programs into countless small programs through the network "cloud" and then processing and analyzing these small programs through a system composed of multiple servers to obtain results and return them to users. Its architecture can be divided into application layer, platform layer, data layer, infrastructure layer, and hardware virtualization layer [14, 15]. Cloud accounting can be understood from two aspects. First, from the perspective of the provider, cloud accounting service is composed of hardware foundation and software foundation. The most important hardware basis is the computer platform, and other types of hardware include servers, network storage, and integrated management system; second, from the perspective of enterprise users, it needs to pay a certain service fee; in the service system, you can enjoy the software processing accounting work service.

2.2. Data Integrity Verification Scheme

2.2.1. Data Integrity Verification Model. In a data integrity verification scheme, according to whether the trusted third party is introduced to verify the data integrity, the system model of the scheme is divided into two types: two-party model and three-party model. Two-party models refer to the model that only verifies data integrity between users and cloud storage server. For the three-party model, the user entrusts the data verification to a trusted third party, and the user only needs to know the verification results [16]. Referring to an object outside of two interrelated subjects, called the third party, the third party can be connected to or independent of the two subjects. Considering that the scheme proposed in this paper introduces the trusted third party, we will focus on the tripartite model.

In the tripartite model, it is generally divided into three parts: user, cloud storage server, and trusted third party. The specific responsibilities of each part are as follows [17, 18]:

(1) User: the user is not only the owner of data but also the purchasing user of the cloud storage service. Users have a lot of data, and the local computing and

storage resources are limited. Users use cloud storage services to reduce the local storage burden. In addition, users can update their own data in cloud storage in real time.

(2) Cloud storage server: storage server with huge storage space can provide users with convenient data storage and data management services, but it is an untrusted organization, which may threaten the integrity of data in the cloud.

(3) Trusted third party: as an agent trusted by users, a trusted third party also has relatively large computing power. Users with limited computing resources can entrust a trusted third party to verify data integrity in cloud storage. However, the trusted third party may be curious about the data that users need to verify so as to pry into the privacy of user data.

As shown in Figure 1, it is a tripartite model of three data integrity verification schemes.

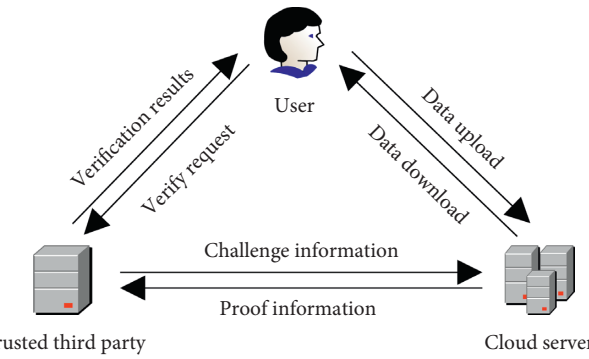

Figure 1: Tripartite model of data integrity verification scheme.

2.2.2. Composition Algorithm of Data Integrity Verification Scheme. In the three-party verification system model, users upload their own data to cloud storage; cloud storage server stores and manages user's data; trusted third party acts as an agent to verify the integrity of data in cloud storage and returns the verification results to users. For a three-party verification system, the data integrity verification scheme generally includes six polynomial time algorithms: system initialization algorithm, key generation algorithm, data label generation algorithm, data integrity verification challenge algorithm, proof generation algorithm, and proof verification algorithm [19].

(1) System initialization: in this stage, input a security parameter k to obtain the initialized system parameter param, which is a probabilistic algorithm and executed by the user.

$$\text{Setup}\left(1^k\right) \longrightarrow \text{param}. \tag{1}$$

(2) Key generation algorithm: in this stage, the system parameter param is input to generate the key pair (pk; sk) required in the data integrity verification process, which is a probabilistic algorithm and executed by the user.

$$\text{KeyGen}\left(\text{param}\right) \longrightarrow \left(\text{pk}, \text{sk}\right). \tag{2}$$

(3) Data label generation algorithm: in this stage, firstly, the data f to be uploaded is partitioned to obtain $F = \{m_1, m_2, \ldots, m_n\}$. Input data block m_i and key sk, and calculate the corresponding data label σ_F, which is a probabilistic algorithm and executed by users.

$$\text{TagGen}\left(F, \text{sk}\right) \longrightarrow \sigma_F. \tag{3}$$

(4) Data integrity verification challenge algorithm: in this stage, the trusted third party initiates a challenge to the cloud storage server and inputs the data name F_{ID} and system parameter param in the cloud storage to generate a chal corresponding to the data challenge information. It is a randomized algorithm and implemented by the trusted third party.

$$\text{Challenge}\left(F_{\text{id}}, \text{param}\right) \longrightarrow \text{chal}. \tag{4}$$

(5) Proof generation algorithm: in this stage, challenge information chal is input, and the cloud storage server generates corresponding data proof P_F and label proof P_σ, which is a probabilistic algorithm and executed by the cloud storage server.

$$\text{ProofGen}\left(\text{chal}, F\right) \longrightarrow \left(P_F, P_\sigma\right). \tag{5}$$

(6) Proof verification algorithm: in this stage, challenge information chal, key pk, data proof P_F, and label proof P_σ are input, and "TRUE" or "FALSE" are output. It is a deterministic algorithm and executed by a trusted third party, where "TRUE" indicates that the data in cloud storage are well preserved; "FALSE" indicates that the data in cloud storage are not well preserved.

$$\text{ProofVer}\left(\text{chal}, \text{pk}, P_F, P_\sigma\right) \longrightarrow \{\text{"TRUE"}, \text{"FALSE"}\}. \tag{6}$$

2.2.3. Security Model of Data Integrity Verification Scheme. For a data integrity verification scheme, we need to prove its security. Generally, the formal definition of scheme security is given by the game model.

In the data integrity verification game model, we can regard the trusted third party as challenger B and the untrusted cloud storage server as adversary A. A data integrity verification game includes the following parts [20, 21]:

(1) Initialization phase: challenger B runs initialization algorithm and key generation algorithm and sends public parameters and public key to adversary A.

(2) Interrogation phase: adversary A selects some data blocks and then sends a query to challenger B about the tags corresponding to these data. Challenger B runs the data label generation algorithm to generate

the corresponding tags for these data blocks and then returns the tags to adversary A.

(3) Challenge stage: challenger B generates a challenge message chal and sends it to opponent A. This challenge information does not include blocks that have been asked in the inquiry phase before.

(4) Verification phase: adversary A tries to forge data proof and label proof according to challenge information and returns the forged certificate to challenger B. If the proof passes the verification of challenger B, opponent A wins the game. Otherwise, it fails.

Through the above security game model, we get the following security definition of data integrity verification scheme: if a data integrity verification scheme is secure, the probability of winning the above game for any opponent with probability polynomial time is negligible, and this probability is equal to the probability of obtaining all the data by using the message collector [22].

2.3. Improvement of Data Integrity Verification Mechanism. In this paper, an improved multibranch path tree authentication structure is proposed and applied to the multitree. It is a balanced tree based on the hash operation characteristics between LBT nodes. The structure uses a hash tree with multibranch paths, and each node (including the root node) adopts the traversal sorting method of increasing numbers from top to bottom and from left to right. By storing data block information on each node, not just on the leaf node, the utilization rate of each node can be improved [23, 24].

2.3.1. System Improvement. LBT is an authentication structure tree with multibranch paths, and its nodes store data block information [25]. Suppose that the user divides the file m into n blocks: $m = (m_1, m_2, \ldots, m_n)$, the out degree of the tree is p, the depth is q, and the LBT structure is constructed. The hash value of a node is obtained by linking the hash value $h(m_i)$ of its corresponding data block with the hash value of child node m_{ix} (where $x \in [1, p]$ and is an integer). The operation formula is shown in

$$f(m_i) = h\left(h(m_i) \| f(m_{i1}) \| f(m_{i2}) \| \ldots \| f\left(m_{ip}\right)\right). \quad (7)$$

If the node has no child node and is a leaf node, its hash value is its own corresponding data block hash value $h(m_i)$:

$$f(m_i) = h(m_i). \quad (8)$$

The auxiliary authentication information is the set of sibling nodes of all nodes in the authentication path, which is denoted as Ω_i.

According to $\{h(m_i), \Omega_i\}$, the auditor first calculates the value of root node R in LBT structure and then compares the calculated value with the previously stored root hash value to detect whether the position of data block is correct, so as to verify the integrity of data in the cloud storage server. If it is consistent, it proves that the data are complete; if it is inconsistent, it means that the data have been destroyed and operations such as addition, deletion, and modification have taken place.

In order to improve the utilization of nodes, shorten the length of the authentication path, and improve the audit efficiency of the audit side, this scheme stores data in each node of the improved multibranch path tree LBT and retains the unique characteristics of hash operation between nodes in traditional MHT. Suppose that the file M is divided into 16 blocks, the out degree of the tree structure is 4, and the depth is 3.

In the improved integrity audit scheme of LBT data structure, it is assumed that the cloud audit side requests to verify the integrity of data blocks m_3 and m_{14}. When the cloud audit side verifies data block m_3, only one hash operation is needed:

$$f(m_1) = h\left(h(m_1) \| f(m_3) \| f(m_2) \| f(m_4) \| f(m_5)\right). \quad (9)$$

The root node r for integrity verification can be obtained. In the same way, m_{14} can be tested with only two hash operations:

$$\begin{aligned} f(m_4) &= h\left(h(m_4) \| f(m_{14}) \| f(m_{15}) \| f(m_{16})\right), \\ f(m_1) &= h\left(h(m_1) \| f(m_4) \| f(m_2) \| f(m_3) \| f(m_5)\right). \end{aligned} \quad (10)$$

2.3.2. Specific Implementation Plan. Bilinear mapping is defined as e: $G \times G \longrightarrow G_T$, where G is a cyclic multiplicative group and is also a Grap Diffie Hellman (GDH) group, while G_T is another cyclic multiplicative group with prime order p. G is a living member of group G and $h(\cdot)$: $\{0, 1\} \longrightarrow G$ is a cryptographic hash function [26, 27]. The user file M is divided into N data blocks: $M = (m_1, m_2, \ldots, m_n)$.

The data integrity audit scheme based on the improved multibranch path LBT structure is divided into three stages, each of which is composed of the following polynomial algorithms.

The initialization stage KeyGen (1^k): cloud audit end randomly selects a number $\alpha \longleftarrow Z_p$, $u_1, u_2, \ldots, u_s \longleftarrow G$ and calculates $v \longleftarrow g^{\alpha}$. Then, the private key of the cloud audit end is $\mathrm{sk} = \alpha$, and the public key is $\mathrm{pk} = \{v, \{u_j\}_{1 \le j \le s}, g\}$.

Upload phase TagGen (M; sk): for each block of file $M = (m_1, m_2, \ldots, m_n)$, the user randomly selects an element $\delta \longleftarrow G$ to make the unique identifier of file M as $\wedge = \mathrm{name}||n||\delta||\mathrm{sig}_{\mathrm{sk}}(\mathrm{name}||n||\delta)$. Then, the client sends the ID file $\wedge$ and the block data m_i $(i = 1, 2, \ldots, n)$ of file M to the TPA of the cloud audit side. After receiving the file, TPA calculates the root node $R\, f(R)^{\alpha} \longleftarrow \mathrm{sig}_{\mathrm{sk}}(f(\mathrm{R}))$ through the improved LBT authentication data structure and signs the root node R with its own private key $\mathrm{sk} = \alpha$. The tag $t = \mathrm{sigsk}(f(R))$ is sent to the client as a message confirmation [28].

After that, TPA will sign each small data block $m_i = (m_{i1}, m_{i2}, \ldots, m_{in})$, where $i = 1, 2, \ldots, n$. The signature algorithm is as follows:

$$\sigma_i \leftarrow \left(h(m_i) \cdot \Pi_J^S = {}_i u_j^{mij}\right)^{\alpha}. \quad (11)$$

The data block signature set $\Phi = \{\sigma_i\}_{1 \le i \le n}$ is obtained. The TPA of the cloud audit side sends the initialization file $m^* =$

$\{M, \Phi, A, t\}$ to the CSS of the cloud storage side and then deletes the local file, and only label t is reserved.

Challenge $(\cdot)$ stage: the authorized auditor randomly selects C elements from the block index set $[1, n]$ to form the data block challenge subset $Q = \{(i, v_i)\}_{1 \le i \le c}$, where $v_i \leftarrow f(t, i, \tau)$ and τ is the timestamp. Then, the generated challenge information pairs are sent to the cloud storage end periodically to complete the verification request task.

The response phase GenProof (M, T, chal, and pk): after cloud storage receives the set of challenge information pairs, it runs an evidence generation algorithm and calculates the following:

$$\mu_j = \sum_{\{(i,v_i)\} \in Q} v_i m_{ij} \in Z_p, \tag{12}$$

where $j = 1, 2, \ldots, s$ and

$$\sigma = \prod_{\{(i,v_i)\} \in Q} \sigma_i^{v_i} \in G. \tag{13}$$

Then it will be the evidence of data integrity

$$p = \left\{ \{\mu_j\}_{1 \le j \le s}, \sigma, \{h(m_i), \Omega_i\}_{1 \le j \le c}, \mathrm{sig}_{\mathrm{sk}}(f(R)) \right\}. \tag{14}$$

It will send it back to the TPA of the cloud audit end.

Audit stage verifyproof (P, pk): cloud audit terminal TPA receives evidence P and runs audit algorithm. First, the root hash value $f(R)$ is calculated by returning $\{h(m_i), \Omega\}_{1 \le i \le c}$ in evidence P and then verifies that $e(t, g) = e(f(R), v)$. If the equation is not true, the verification fails, and reject is output. If and only if the $f(R)$ verification passes, the equation continues to be verified

$$e(\sigma, g) = e\left(\prod_{\{(i,v_i)\} \in Q} h(m_i)^{v_i} \cdot \prod_{j=1}^{s} u_j^{\mu_j}, v \right). \tag{15}$$

If yes, the system outputs accept to prove that the data is complete; if not, reject is output, indicating that the data have been destroyed and operations such as addition, deletion, and modification have been sent.

3. Simulation Experiment of Cloud Data Integrity Verification Algorithm

In the performance analysis of the data integrity proof mechanism proposed in this paper, the scheme is mainly compared with the other two data integrity mechanisms. In the time analysis, it mainly analyzes the comparison between the time of constructing the scheme data structure and the other two data structures, the time comparison of dynamic operation, and the change of the evidence generation time of the server, and the evidence verification time of the third-party verifier when the scheme implements the data integrity proof mechanism.

3.1. Experimental Simulation Environment. In the process of experimental simulation, one computer will be used to simulate the cloud audit end and the other computer to simulate the cloud storage end. Under the 64-bit windows 10 operating system, this scheme and the other two schemes are implemented based on Java language, and the performance gap of the three schemes is compared. The hardware parameters are Intel Core i7 processor, 8 GB memory, 256 g SSD, and 2.5 GHz CPU. The simulation software eclipse 2012 is used. The challenge block number I is selected by pseudorandom $f(x) = \mathrm{rand}(\cdot)$. All the simulation results are the average of 50 experiments under the same experimental conditions.

3.2. Experimental Simulation Object. As shown in Table 1, the basic performance of the other two classical data integrity proof schemes is given. By analyzing the performance of the scheme proposed in this paper, the performance comparison table between the scheme proposed in this paper and other typical schemes is given.

4. Comparison of Cloud Data Integrity Verification Algorithms for Accounting Informatization

4.1. Cloud Server Computing Time Comparison. As shown in Table 2 and Figure 2, the results show that the greater the number of data blocks, the greater the difference in computing efficiency of each algorithm; in the case of the same number of data blocks, the calculation efficiency of the algorithm in this paper is better than the other two algorithms; when the algorithm output is larger, the computing time of the cloud server is shorter.

4.2. Operation Time of Data Block Changes. As shown in Table 3, the 100 m size file is partitioned according to 1 KB, and the data integrity proof mechanism of the three algorithms performs data block insertion, update, and deletion operations and compares the time change when updating different number of data blocks.

4.2.1. Insert Data Block. As shown in Figure 3, the data integrity proof mechanism proposed in this paper can update, delete, and add dynamic operations of continuous data blocks at one time when performing dynamic operations. However, in the other two data integrity proof mechanisms, if you want to insert multiple data blocks in one location, you can only insert one data block at a time and repeat the operation until all the data blocks are inserted into the data file. The execution time of the other two schemes increases with the increase of data blocks, while the execution time of this scheme remains unchanged.

4.2.2. Update Data Block. As shown in Figure 4, the data block update operation is performed for three kinds of data integrity proof mechanisms, and the time change of updating different number of data blocks is compared. The execution time of scheme 1 and scheme 2 increases with the increase of data blocks, while the execution time of this scheme remains basically unchanged. In the early stage of

Table 1: Performance comparison of different schemes.

Program	Program 1	Program 2	This program
Public verification	No	Yes	Yes
Dynamic update	Yes	Yes	Yes
Security	Yes	Yes	Yes
Batch audit	No	No	Yes
Verifier computing overhead	$O(c)$	$O(c)$	$O(\log n)$
Communication overhead	$O(\log n)$	$O(\log n)$	$O(\log n)$
Server computing overhead	$O(n)+O(\log n)$	$O(\log n)$	$O(\log n)$
Client computing overhead	$O(\log n)$	$O(\log n)$	$O(\log_k n)$

Table 2: Cloud server computing time change.

Number of blocks	50	100	200	350	500
Program 1	50	81	175	334	432
Program 2	50	79	161	276	403
This program (out degree 8)	49	64	139	254	382
This program (out degree 16)	49	64	139	238	347

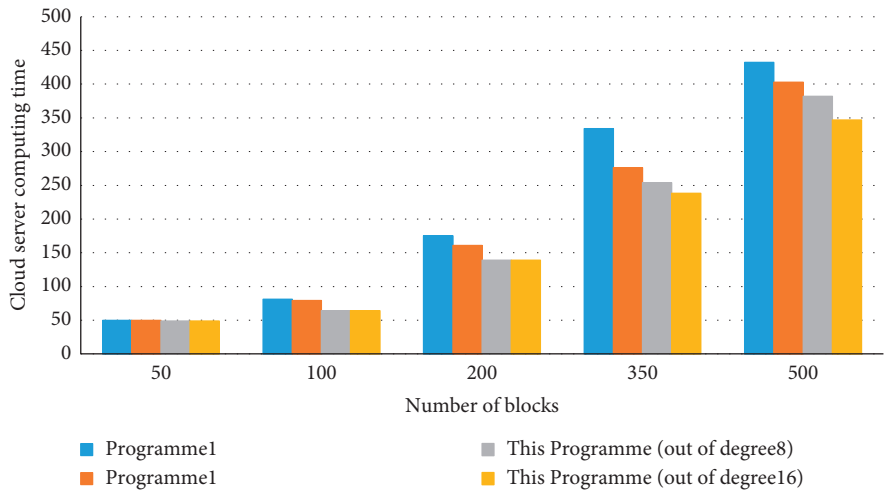

Figure 2: Cloud server computing time change chart.

Table 3: Comparison of addition, deletion, and modification time of three algorithms.

		2	4	6	10	16
Insert	Program 1	58	74	152	253	371
	Program 2	43	61	98	136	197
	This program	26	24	25	27	26
Update	Program 1	19	28	34	32	46
	Program 2	16	22	27	31	35
	This program	11	10	10	12	11
Delete	Program 1	66	104	152	281	372
	Program 2	42	78	103	179	234
	This program	25	28	24	23	25

the experiment, the difference between the three methods was not very obvious. At the first four minutes, the gap began to widen, and at the tenth minute, the gap between the first two groups of algorithms narrowed.

4.2.3. Delete Data Block. As shown in Figure 5, the data block deletion operation is performed for the three data integrity proof mechanisms, and the time change of updating different number of data blocks is compared. The execution time of scheme 1 and scheme 2 increases with the increase of data blocks, while the execution time of this scheme remains basically unchanged.

4.3. Communication Cost Comparison. In this paper, we simulate the communication overhead generated in the challenge response phase and only compare the

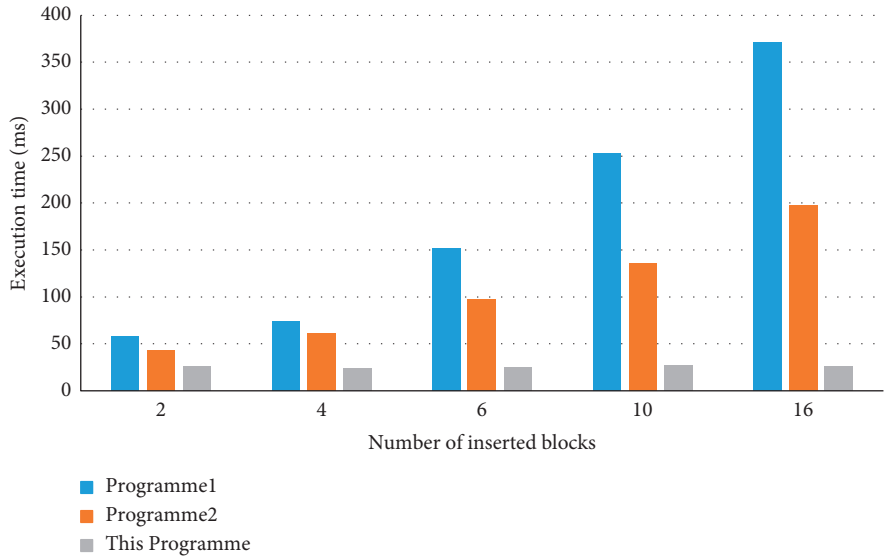

Figure 3: Comparison of execution time of inserting data block among three algorithms.

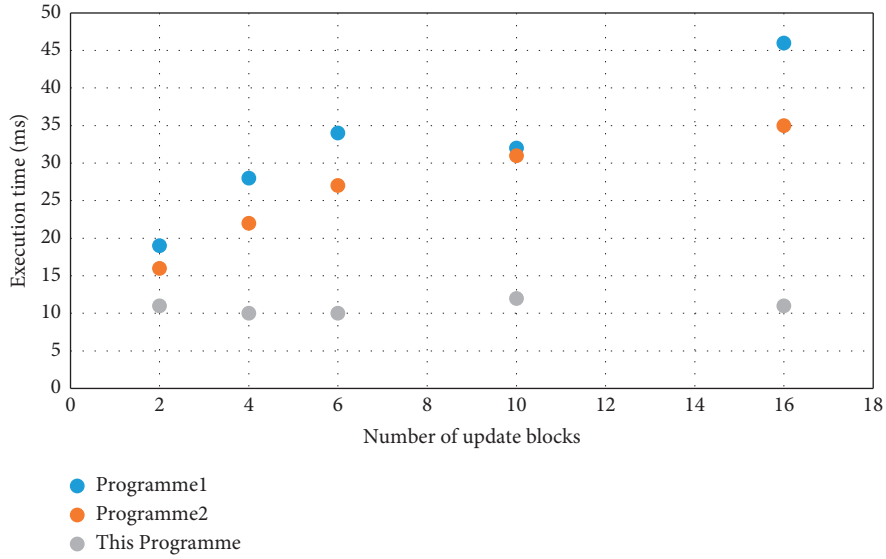

Figure 4: Update data block time change comparison.

communication overhead when challenging a single data block each time. The communication cost of batch processing is similar to that of a single data block and increases to its multiple. Challenge response means that the user sends a password to the remote host. The remote host sends the user a challenge message (encrypted information) according to the password. The user generates a response message based on his password and the corresponding algorithm to match the challenge message. If the match is successful, the authentication is successful; if the match fails, the authentication fails.

As shown in Table 4 and Figure 6, it shows the data size relationship of interaction between cloud audit TPA and cloud storage side CSS when using a single data block of different sizes. When the data block size is 10 KB, the communication cost of scheme 1 is 0.024 KB, that of scheme 2 is 0.028 KB, and that of this scheme is 0.027 KB. Compared with scheme 1, the authentication path of scheme 1 is shorter than that of scheme 1, and the communication cost is slightly higher after multiple weighting; compared with scheme 2, the communication cost of this scheme is slightly smaller, but the difference is not significant.

4.4. Comparison of Computing Costs. In the challenge response phase, this paper mainly analyzes the computing cost of TPA on the integrity evidence P returned by CSS of cloud storage side, including the computing cost caused by retrieving data authentication tree structure node, root node R, and label generation algorithm.

As shown in Table 5 and Figure 7, the relationship of audit time used by TPA of cloud audit side and the audit time of different number of data blocks in a batch is listed.

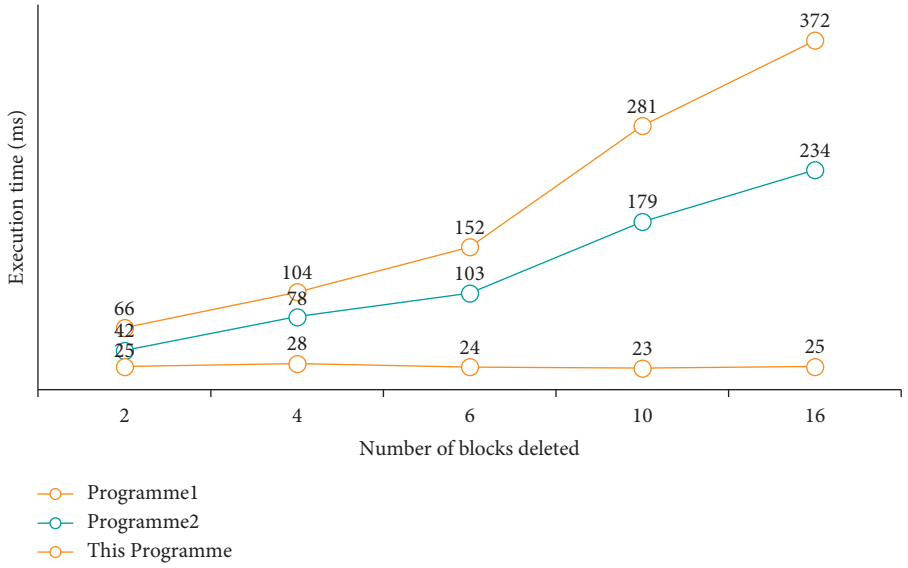

Figure 5: Delete block execution time comparison chart.

Table 4: Communication cost comparison of three schemes.

Block size (k)	5	10	15	20	25
Program 1	0.018	0.023	0.024	0.034	0.038
Program 2	0.018	0.021	0.028	0.035	0.041
This program	0.018	0.023	0.027	0.033	0.040

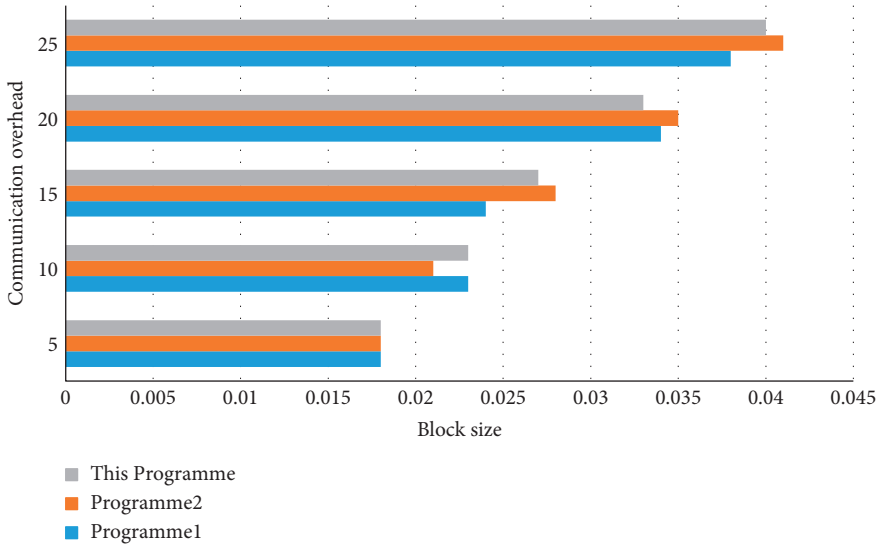

Figure 6: Communication cost comparison of three schemes.

Table 5: Single time comparison.

Block size (k)	10	20	30	40	50
Program 1	0.07	0.15	0.21	0.32	0.42
Program 2	0.07	0.14	0.20	0.27	0.41
This program	0.07	0.14	0.18	0.27	0.36

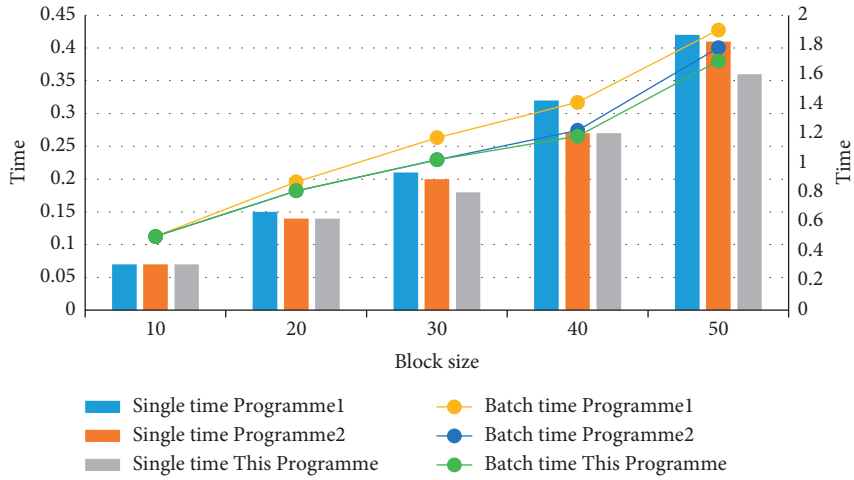

Figure 7: Comparison of computing costs.

For example, in the audit efficiency of a single data block, when the size of data block is 30 KB, the time required for scheme 1 to calculate root node R is 0.21 s, that of scheme 2 is 0.20 s, and that of this scheme is only 0.18 s. From the above results, it can be seen that, compared with the other two schemes, the audit speed of this scheme is faster, and it takes less time. This is because the multibranch structure is adopted in this scheme. When the number of target data blocks is the same, only a shorter layer of authentication tree is needed to cover all data blocks, which shortens the length of the authentication path and shortens the time of calculating root node R.

5. Conclusions

With the continuous popularity of the Internet and mobile devices, networked storage will become the main way of storage in the future, and cloud storage will also be the inevitable trend of networked storage. More and more users will choose cloud storage. While users experience convenient storage, they also lose the direct control of files. The security of data on the cloud server has been tested. In order to maintain the user's control of cloud files, data integrity certification emerges as the times require. Data integrity certification in cloud storage environment has attracted many researchers, and integrity proof has become one of the research hotspots with the development of cloud storage.

Firstly, this paper studies the current development of data integrity proof mechanism, including the basic model of data integrity proof, including system model, security model, and PDP and POR, two basic models commonly used in integrity proof. It introduces the main algorithm and implementation process used in the two basic models and analyzes the characteristics and shortcomings of the existing schemes. The correctness, security, and performance tests show that the scheme is feasible.

Although this paper improves the data update, security, and performance of the scheme on the basis of the existing scheme, we can also do the following further research and improvement on the data integrity scheme in the future work: for multiple dynamic update operations, the authentication structure tree will become very uncoordinated and need to be reconstructed. Therefore, we hope to find a data authentication algorithm that needs only a little modification and no reconstruction even after dynamic update operations.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The author declares that there are no conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] S. Namasudra and P. Roy, "PPBAC: popularity based access control model for cloud computing," *Journal of Organizational and End User Computing*, vol. 30, no. 4, pp. 14–31, 2018.

[2] W. Amol and V. Rastogi, "Data integrity auditing of cloud storage," *International Journal of Computer Applications*, vol. 133, no. 17, pp. 17–21, 2016.

[3] L. Zhou, V. Varadharajan, and K. Gopinath, "A secure role-based cloud storage system for encrypted patient-centric health records," *The Computer Journal*, vol. 59, no. 11, pp. 1593–1611, 2016.

[4] G. Chen, Y. Lu, Y. Meng et al., "FUSO: fast multi-path loss recovery for data center networks," *IEEE/ACM Transactions on Networking*, vol. 26, pp. 1376–1389, 2018.

[5] A. F. Barsoum and M. A. Hasan, "Provable multicopy dynamic data possession in cloud computing systems," *IEEE Transactions on Information Forensics & Security*, vol. 10, no. 3, pp. 485–497, 2017.

[6] Y. Zeng, G. Chen, K. Li, Y. Zhou, X. Zhou, and K. Li, "M-skyline: taking sunk cost and alternative recommendation in consideration for skyline query on uncertain data," *Knowledge-Based Systems*, vol. 163, no. 1, pp. 204–213, 2019.

[7] K. Omote and P.-T. Tran, "D2-POR: direct repair and dynamic operations in network coding-based proof of

retrievability," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 4, pp. 816–829, 2016.

[8] L. Monnerat and C. L. Amorim, "An effective single-hop distributed hash table with high lookup performance and low traffic overhead," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 7, pp. 1767–1788, 2015.

[9] C. Clune, "Accounting and networks of corruption," *Social and Environmental Accountability Journal*, vol. 35, no. 1, pp. 67-68, 2015.

[10] B. Lee, N. Murray, and Y. Qiao, "Active accounting and charging for programmable wireless networks," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 111–120, 2015.

[11] B. A. Rutherford and D. Northcott, "The struggle to fabricate accounting narrative obfuscation: an actor-network-theoretic analysis of a failing project," *Qualitative Research in Accounting & Management*, vol. 13, no. 1, pp. 57–85, 2016.

[12] E. V. B. Murro and I. M. Beuren, "Actor networks in specialized accounting inspection: an analysis in the light of the actor-network theory," *Review of Business Management*, vol. 18, no. 62, pp. 633–657, 2016.

[13] V. Pinsky and A. Shapira, "Seismic network detection probability assessment using waveforms and accounting to event association logic," *Journal of Seismology*, vol. 21, no. 1, pp. 69–82, 2017.

[14] W.-C. Wu and H.-T. Liaw, "An authentication, authorization, and accounting mechanism for 3G/WLAN networks," *Security and Communication Networks*, vol. 9, no. 6, pp. 468–480, 2016.

[15] K. L. Bills, C. Hayne, and S. E. Stein, "A field study on small accounting firm membership in associations and networks: implications for audit quality," *The Accounting Review*, vol. 93, no. 5, pp. 73–96, 2018.

[16] C. Lin, Z. Shen, Q. Chen, and F. T. Sheldon, "A data integrity verification scheme in mobile cloud computing," *Journal of Network and Computer Applications*, vol. 77, pp. 146–151, 2017.

[17] X. Wang, Y. Lin, and G. Yao, "Data integrity verification scheme with designated verifiers for dynamic outsourced databases," *Security & Communication Networks*, vol. 7, no. 12, pp. 2293–2301, 2015.

[18] A. Alsharif, M. Nabil, S. Tonyali, H. Mohammed, M. Mahmoud, and K. Akkaya, "EPIC: efficient privacy-preserving scheme with EtoE data integrity and authenticity for AMI networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3309–3321, 2019.

[19] S. H. Abbdal, T. A. Kadhim, Z. A. Abduljabbar et al., "Ensuring data integrity scheme based on digital signature and Iris features in cloud," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2, no. 2, p. 452, 2016.

[20] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1717–1726, 2015.

[21] B. Haiyong and C. Le, "A lightweight privacy-preserving scheme with data integrity for smart grid communications," *Concurrency & Computation Practice & Experience*, vol. 28, no. 4, pp. 1094–1110, 2016.

[22] T. Bardini Idalino, L. Moura, R. F. Custódio, and D. Panario, "Locating modifications in signed data for partial data integrity," *Information Processing Letters*, vol. 115, no. 10, pp. 731–737, 2015.

[23] T. Halevi, N. Saxena, and S. Halevi, "Tree-based HB protocols for privacy-preserving authentication of RFID tags," *Journal of Computer Security*, vol. 19, no. 2, pp. 343–363, 2016.

[24] S. Vig, R. Juneja, G. Jiang, S.-K. Lam, and C. Ou, "Framework for fast memory authentication using dynamically skewed integrity tree," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 10, pp. 2331–2343, 2019.

[25] M. Kim, W.-K. Choi, and M.-S. Jun, "A design of efficient multi-authentication scheme using a merkle hash tree in the smart home environments," *Advanced Science Letters*, vol. 22, no. 9, pp. 2538–2542, 2016.

[26] H. Zang, Y. Huang, H. Cao, and C. Li, "A novel privacy protection protocol for vehicular ad hoc networks based on elliptic curve bilinear mapping," *Ingénierie des Systèmes d Information*, vol. 24, no. 4, pp. 397–402, 2019.

[27] M. Kech and F. Krahmer, "Optimal injectivity conditions for bilinear inverse problems with applications to identifiability of deconvolution problems," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 20–37, 2017.

[28] S. Cheng, Z. Yadong, W. Lei, and L. Zhizhong, "Research on k-anonymity privacy protection scheme based on bilinear pairings," *The Journal of China Universities of Posts and Telecommunications*, vol. 25, no. 5, pp. 18–25, 2018.