Hindawi
Security and Communication Networks
Volume 2022, Article ID 5016809, 15 pages
https://doi.org/10.1155/2022/5016809

Research Article

An Edge Cloud Data Integrity Protection Scheme Based on Blockchain

Weihua Duan,[1] **Yu Jiang**,[1] **Xiaolong Xu**,[1,2] **Ziming Zhang**,[1] **and Guanpei Liu**[1]

[1]*Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*

Correspondence should be addressed to Xiaolong Xu; xuxl@njupt.edu.cn

Received 1 January 2022; Revised 20 March 2022; Accepted 1 April 2022; Published 22 April 2022

Academic Editor: Xin-Yi Huang

The publicly accessible feature of edge servers leads to the threat of malicious access to the data stored on the server and a series of security problems such as the leakage of user data privacy and the destruction of integrity. Data custody causes the separation of user ownership and management rights and brings potential security risks of data theft and destruction. Among them, for the integrity of the data uploaded by the terminal, the current protection mechanism mostly verifies the identity of the visitor or encrypts the data, but the role of verification is mostly assumed by the server, and it is impossible to avoid the collusion of edge servers with malicious intruders. In this paper, a distributed virtual machine agent (VMA) is designed and implemented, an edge cloud data integrity monitoring framework is built, and the verification protocol based on blockchain is proposed, which achieves trusted verification without relying on a trusted third party. Also, a prototype system of edge cloud data integrity protection based on blockchain is constructed to prevent data corruption. The results of security proof and experimental verification show that the mechanism based on blockchain technology can defend against three attacks of cloud service providers, has superior computation, and reduces the storage costs to protect the integrity of user data.

1. Introduction

Cloud computing [1–3] is a computing model that uses the Internet anytime, anywhere, and quickly access shared resource pools (such as computing facilities, storage devices, and applications) in the form of on-demand services to provide users. However, with the increasing number of devices in the network and the exponential growth of generated data, cloud computing is difficult to handle massive amounts of business, resulting in a large amount of time delay, providing users with unsatisfied service experience. Therefore, edge computing [3, 4] is proposed to provide services that are closed to IoT devices with a short delay. An edge cloud [2, 5, 6] consists of edge servers located close so that it can use server collaboration to complete tasks from the IoT devices more efficiently and realize the real-time need.

Edge devices depending on distributed edge servers that belong to different enterprises and suppliers are scattered and their processing capabilities are limited. Due to the heterogeneity of devices, most authentication and communication encryption technologies are not suitable for networking, causing data to suffer a huge threat, and data integrity cannot be guaranteed. Data custody causes the separation of user ownership and management rights and brings potential security risks of data theft and destruction.

On the one hand, the Edge Cloud Service Provider (ECSP) may either privately delete user data or deliberately conceal accidental data destruction for maintaining its own reputation. On the other hand, edge servers may be maliciously attacked, resulting in data destruction and loss of sensitive data. Edge cloud data integrity protection mechanism can ensure that data are stored in the edge cloud unmistakably and can immediately warn and reduce losses when data are illegally tampered.

The traditional methods of data integrity verification mainly focus on the integrity of local disk data and database data and adopt the scheme of integrity verification such as

digital signature, message verification code, and digital watermarking [7], but the current data integrity research is mainly for data in cloud computing. Yan et al. [8] adopted a combination monitor of inside and outside in a virtual machine, which proposed the security proposal for a virtual machine computing environment. The method provides the monitoring architecture of the virtual machine to ensure trusted computing but increases the hardware cost of the cloud platform and extra computing costs, especially since the architecture is not scalable. Erway et al. [9] improved a PDP scheme based on a Rank-based Authenticated Skip List (RASL). In 2015, Tian et al. [10] provided a dynamic data integrity mechanism based on the distributed hash table (DHT), which supports public authentication. However, the verification is conducted by the third-party auditors in these schemes. Users may be deceived by fake verification and collusion with CSP. Xu et al. [11] proposed a data validation algorithm to defend against spoofing attacks from untrusted validation results, which improved the reliability of validation results in 2017. However, dual validation evidence was introduced to cross-validate the validation results, which added computation and storage costs.

In edge computing, each server can process tasks for users independently and save the uploaded data. When the tasks submitted are difficult to handle, they will be submitted to the cloud center. This kind of scene realizes the decentralized scene at the edge and refuses the centralized manager.

The research on data integrity protection of edge cloud has made some progress in recent years. Wang et al. [12] achieved a balance through a balanced truth discovery method and the proposed data privacy enhancement technology and used these technologies to interact with IoT devices and edge servers. Chadwick et al. [13] proposed a framework that allows the secret sharing of cyber threat information (CTI) among partners for analysis. Li et al. [14] proposed a privacy protection data aggregation scheme for mobile edge computing-assisted IoT applications. The data privacy of the terminal device is guaranteed, and source authentication and integrity are also provided. Wang et al. [15] proposed an edge-based data collection model, in which the raw data from the wireless sensor network (WSN) is differentially processed by an algorithm on the edge server for privacy calculations.

Recently, Blockchain technology [16–18] has become popular worldwide. The blockchain guarantees the consistency of the data between nodes by a consensus algorithm and ensures data security by the encryption algorithm. In addition, formed by the timestamp and hash algorithm, the chained structure produces a series of technical features, such as openness, transparency, authentication, and tamper resistance [19]. The theory of smart contracts [20], firstly proposed by Nick Szabo, refers to a computer program which conducts terms of contract automatically. Blockchain technology, with a characteristic of multistorage, multiparty calculation, transparent rules, and tamper-resistant features, provides a reliable record carrier and execution environment for the smart contract.

This paper proposes a distributed virtual machine proxy architecture and a multitenant jointly safeguards the private chain based on the blockchain in the edge cloud and designs an edge cloud data integrity protection mechanism. The mechanism is oriented to the incredible edge cloud system and reaches a consensus agreement to complete credible integrity verification through the exchange of information. The main contributions of this paper are as follows:

1 Mobile Agent is used to deploy the distributed model of the virtual machine agent in the edge cloud. Virtual machine agents of multitenants cooperate to ensure data credible verification. The virtual machine agent mechanism completes not only reliable storage, monitoring, and verification of cloud data tasks but also is necessary to build a data integrity verification mechanism based on blockchain.

2 A blockchain integrity monitoring framework is built through the model of a virtual machine agent. This paper uses the Merkle Hash Tree to generate the unique value corresponding to data and monitor data changes with a smart contract in the blockchain for sending timely warnings of data destruction to the owner. In addition, the "challenge-response" model is used to construct the scheme of edge cloud data integrity verification.

3 This paper constructs and implements a prototype system of edge cloud data integrity protection based on blockchain and applies the integrity monitoring scheme based on virtual machine agents and the integrity verification scheme based on blockchain. After security certification analysis, the mechanism can defend against three kinds of attacks by edge cloud service providers and has a better performance compared with existing solutions.

The rest of the paper is organized as follows. Section 2 introduces the related work about the integrity verification mechanism based on the third party and blockchain technology. Section 3 puts forward blockchain architecture for cloud data integrity based on distributed virtual machine agents. Section 4 presents safety certification according to the scheme. Section 5 perfects experimental verification and performance analysis. Section 6 realizes the prototype system. Finally, section 7 summarizes and evaluates all of the work and points out the direction of further study.

2. Related Work

Data integrity verification in the edge computing environment has attracted more and more scholars' attention. Wang et al. [12] proposed a scheme that maintains a balance in three aspects, including user privacy, data integrity in edge-assisted IoT devices, and computing cost. Through the identity verification algorithm based on biometric ECC, the privacy participation of IoT users is authenticated during the truth discovery process, not only reducing the overall computing cost of the IoT equipment but also limiting the communication between the user equipment and the edge

server. Chadwick et al. [13] proposed a five-level trust model based on cloud edge data sharing infrastructure. Data owners can choose the appropriate level of trust and CTI data cleaning methods, from plain text to anonymization/pseudonymization to homomorphic encryption, so that CTI data can be manipulated before sharing it for analysis. Li et al. [14] proposed a privacy protection data aggregation scheme for mobile edge computing-assisted IoT applications. In the proposed model, there are three participants, namely terminal devices, edge servers, and public cloud centers. The data generated by the terminal device is encrypted and transmitted to the edge server. The edge server aggregates the data of the terminal device and submits the aggregated data to the public cloud center. Finally, the aggregated plaintext data can be recovered by the private key of the public cloud center.

Blockchain technology has been widely used in cryptocurrency since the emergence of Bitcoin [16]. IBM Blockchain [21] offers developers opportunities to develop their own applications based on the Hyperledger Fabric, which has been widely used in the financial industry, insurance industry, food safety, and so on. For instance, IBM and Wal-Mart cooperate to guarantee food safety by food traceability. Azure Blockchain [22] allows customers to quickly configure and deploy consortium chain networks, which supports lightweight development and testing workloads and even large-scale production blockchain deployment. The blockchain can shorten development time and costs through the cloud services required for application development. Amazon Managed Blockchain [23], which helps users use Ethereum and Hyperledger Fabric to create and manage a scalable blockchain network, eliminating the need to create a network, and continuously monitoring the blockchain network to quickly adapt to changes for application requirements has been used in many fields, such as financial and trade alliances. All parties in the blockchain can trade electronically and process trade-related paperwork without central trust.

Wang et al. [15] proposed an edge-based data collection model, in which the raw data from the wireless sensor network (WSN) is differentially processed by an algorithm on the edge server for privacy calculations. A small amount of core data are stored on the edge and local servers, while the rest is transmitted to the cloud for storage. Tian et al. [24] proposed an effective privacy protection authentication framework. By using a lightweight online/offline signature design, authentication efficiency is guaranteed when deployed on small drones with limited resources. Considering the high mobility of drones, a predictive authentication method is studied using mobile edge computing (MEC) in the framework to further reduce the cost of identity verification for potential identity verification activities. In addition, Wang et al. [25] designed a service selection method which selects corresponding credible and reliable service providers based on trust evaluation and recording standards, which has obvious advantages in terms of concise trust management, convenient service search, and accurate service matching. Establishing and maintaining a unified and trusted environment based on edge computing can detect malicious service providers and service consumers in a timely manner, filter out false information, and recommend trusted service providers.

Yue et al. [26] proposed a blockchain-based framework without third-party auditors for data integrity verification in distributed edge cloud storage (ECS) scenarios. In the framework, a Merkle tree with random challenge numbers is used for data integrity verification, and different Merkle tree structures are analyzed to optimize system performance. In view of the problems of limited resources and high real-time requirements, sampling verification is further proposed, and reasonable sampling strategies are formulated to make sampling verification more effective.

Bonnah et al. [27] proposed a completely decentralized method to solve the untrustworthy problem of trusted parties by eliminating the public trusted entity in the network framework. Within the proposed framework, authenticated users do not have to log in to each service provider to be authenticated to access services or resources.

Ma et al. [28] proposed a blockchain-based edge computing trusted data management scheme for dishonest data. They proposed a flexible and configurable blockchain architecture, including mutual authentication protocols, flexible consensus and smart contracts, block and transaction data management, blockchain node management, and deployment. Before data storage in the blockchain system, a user-defined encryption method for sensitive data is designed, and conditional access and decryption queries for protected blockchain data and transactions from the blockchain system are designed.

Kang et al. [29] used blockchain and smart contract technology to realize secure data storage and sharing in the vehicle edge network. These technologies effectively prevent unauthorized data sharing. It also proposed a reputation-based data sharing program to ensure high-quality data sharing between vehicles. A three-weight subjective logic model is used to accurately manage the reputation of the vehicle.

Gai et al. [30] proposed a new method that combines the IoT with edge computing and blockchain. The proposed model is designed for a scalable and controllable IoT system, making full use of the advantages of edge computing and blockchain to establish a privacy protection mechanism while taking into account other constraints, such as energy costs.

In order to efficiently audit the integrity of application vendors' cached data, Li et al. [31] analyzed the threat model and audit objectives and proposed a lightweight sampling-based probabilistic method, including a variable Merkle hash tree. A new data structure of variable Merkle hash trees is designed to implement integrity proofs for generating copies of these data during audits.

Tong et al. [32] proposed two integrity checking protocols for mobile edge computing, checking the data integrity at the edge based on the concept of provable data ownership and proprietary information retrieval techniques. Liu et al. [33] modeled data failures by classifying them into format failures, time series failures, and value failures and proposed several heuristic rules for the detection and

isolation of data failures. Aujla et al. [34] designed a blockchain-based secure data processing framework for the Internet of Vehicles in the edge environment, including a container-based optimal data processing solution and a blockchain-based data integrity management solution, which can minimize link interruptions.

3. Study on Edge Cloud Data Integrity Protection Mechanism

Aiming at the problem of the untrustworthiness of data integrity verification in edge cloud, this paper designs a distributed virtual machine agent model in edge cloud combined with characteristics of blockchain technology and achieves consensus through multinode collaboration to complete the credible verification of edge cloud data. The design focuses on solving three problems. First, the integrity verification by a virtual machine agents prevents data leakage to third-party auditors; Second, credible proof on the blockchain ensures the credible validation results. Third, blockchain monitors the entire lifecycle of user data to ensure that data is not illegally tampered with.

3.1. Distributed Virtual Machine Agent Model. Virtual machine node is divided into two categories in function, including virtual machine agent (VMA) Node and storage node. When the user submits a storage task, the data are preprocessed by the VMA node, which is responsible for selecting the appropriate storage node, and after all the storage is done, the VMA node returns the result to the user. Different from cloud computing, edge computing is to sink resources near the data source and process the user's tasks close to the device. The data does not have to be uploaded to the data center, thus reducing the pressure on network bandwidth. The edge cloud can form a server cluster of edge servers with similar geographical locations and use server cooperation to complete tasks at the edge, reducing the delay of data transmission. Therefore, compared to cloud computing, edge computing is a highly decentralized distributed computing architecture.

For the application of complex services in the edge cloud distributed environment, and to enhance the portability of the model, the paper refers to the cloud environment and uses the Mobile Agent [35, 36] (MA) technology. MA is an agent in the network which performs specific processing in distributed problems. In the standard of FIPA [37] (Foundation of Intelligent Physical Agents), Agency is a container for carrying MA, and it may carry a plurality of MA and provide an operating environment for performing any MA. An agency can carry a number of MA, and MA can be run in the agency. Therefore, the running agency in the node can complete the model deployment of distributed virtual machines agent. Figure 1 is a node structure of the user in the edge cloud.

Definition 1. VMA node, proxy node in edge cloud, logically unique, is responsible for acting on behalf of the user to perform various tasks with high computing power.

Definition 2. Storage node, storage for edge data, not unique. All storage nodes consist of Interplanetary File System [38] (Interplanetary File System, IPFS) cluster which is responsible for storing massive data with lower computing power.

After the deployment of the virtual machine agent model, the paper uses blockchain technology to union nodes, which aims at achieving a consensus agreement through the exchange of information to ensure chain data is open, transparent, tamper-resistant, and traceable. Blockchain is divided into a public chain, private chain, and consortium chain in accordance with the authority of the consensus process. This paper adopts a private chain, giving cloud tenants the privilege to read and write, preventing outside interference in the consensus process. In addition, in order to prevent malicious attacks, we take tokens way to produce a transaction. Each node has a certain initial token, and every deal needs to consume tokens. Once successfully obtained the right to package block, edge nodes will receive some token reward so as to encourage tenants open owner VMA to participate consensus process.

As shown in Figure 2, this paper introduces a distributed virtual machine agent model to build a basic protection framework for edge cloud data integrity. When the user submits the storage task, the data is first uploaded to the VMA node, and after the preprocessing, a transaction is generated into the buffer pool. The transaction stores the evidence of data integrity verification. VMA nodes perform polling, querying the transaction that has not been confirmed in the buffer pool and once found, the VMA is trying to verify the legitimacy of the transaction and packages to form a group of the block legitimate transactions.

3.2. Workflow. The aim of the section is to build a blockchain network through interaction with the VMA for the preparation of integrity protection.

3.2.1. Connection and Synchronization. Blockchain network is based on P2P protocol and there are no central authority nodes. Each node can broadcast routing, discover new nodes, and allow dynamic legitimate nodes to join or quit. The underlying blockchain platform is not limited to Ethernet Square, Ethermint, Fabric, and so on, as long as there are many functions such as account inquiries, transactions, contracts, and other operational intelligence functions.

Step 1. First, the ECSP deploys a blockchain network in the edge cloud and runs the initial file to generate a first block (block Genesis), waiting for the VMA of tenants.

Step 2. Once joining the blockchain network, user's VMA verifies itself whether the data block is the latest in the blockchain network or not. If yes, VMA monitors data broadcasting in the network. Otherwise, block data synchronization neighbor nodes. Then use the public key to verify the legitimacy of transactions.

Step 3. When listening to new transactions and blocks, VMA verifies the signatures of those transactions and

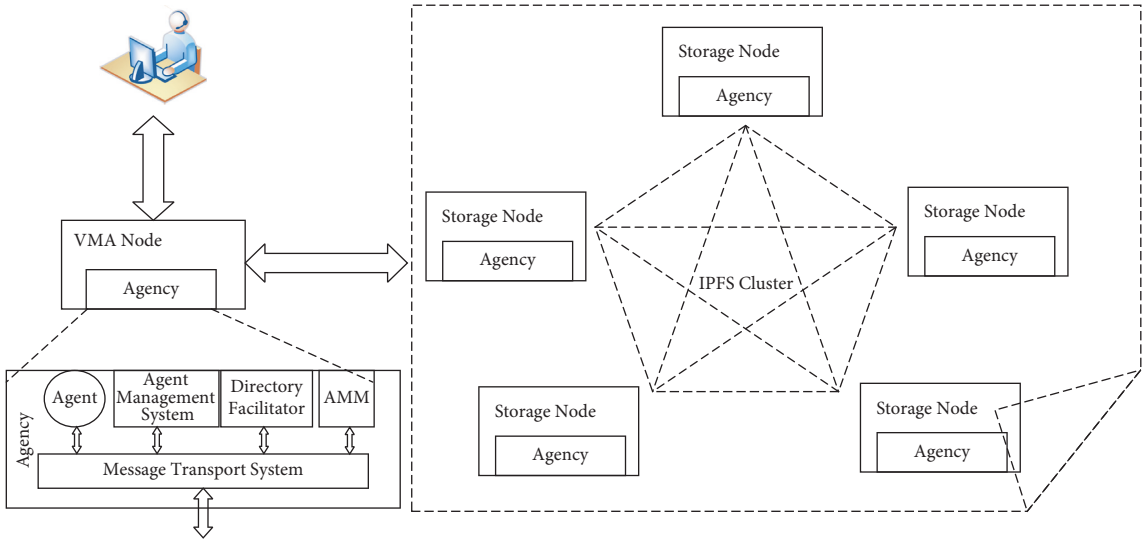

Figure 1: Storage node in edge cloud architecture.

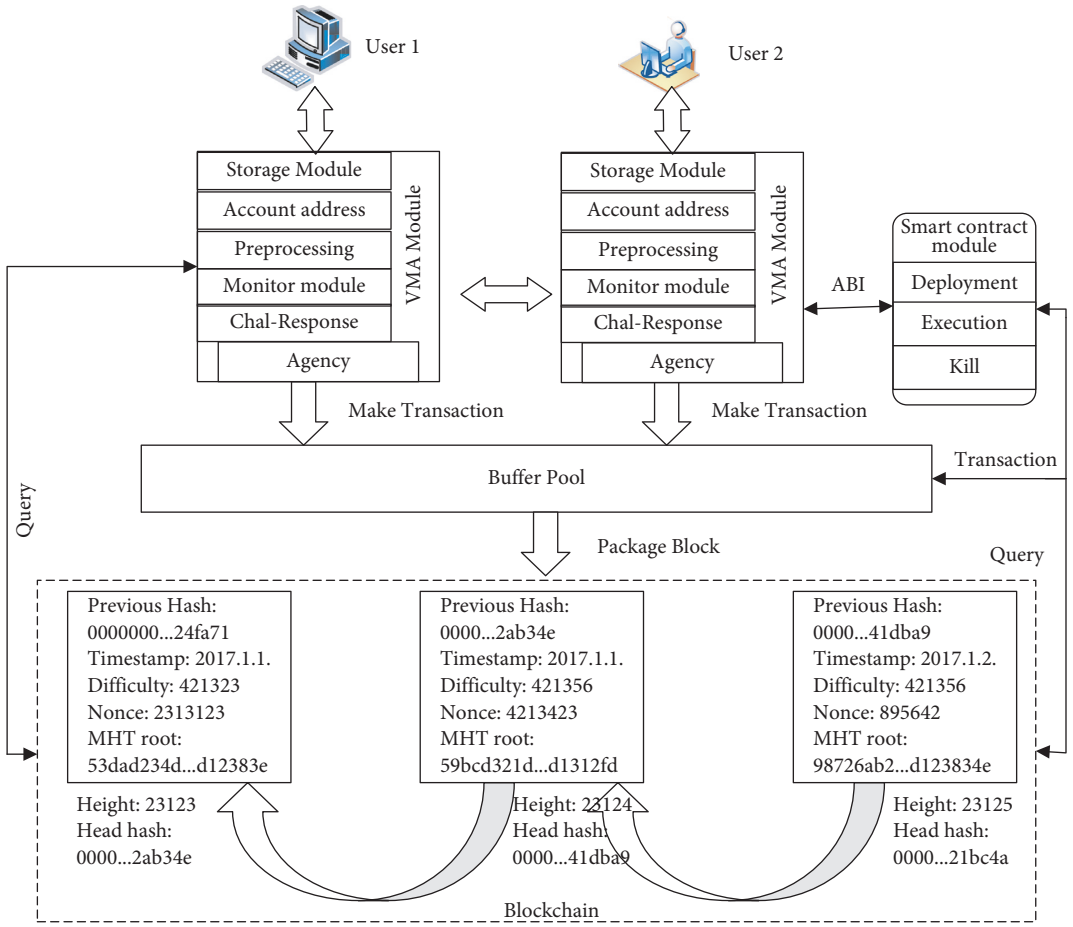

Figure 2: Blockchain architecture for edge cloud data integrity.

blocks. If signatures are valid, VMA processes and forwards them by the consensus module to prevent invalid data from propagating.

3.2.2. Storage. Once users upload files, VMA preprocesses the file and uses IPFS cluster storage. Based on the contents alternate instead of domain, IPFS uses http browser to search files, firstly locates the server, and then uses the pathname to find files on the server. Specific steps are as follows:

Step 1. When file is added to an IPFS node, a unique encrypted hash fingerprint is calculated from the file contents, ensuring that the value always only indicates the contents of the file. Even if you modify a bit of data in a file, the hash fingerprint will be completely different.

Step 2. The next step, users query hashing by the distributed hash table in the IPFS distributed network, which uses a consistent hash function to unify the machine's IP address and data, and quickly (The network only needs 20 hops in a system with 10,000,000 nodes) find the node that owns the data, retrieve the data, and use hashes to verify if this is the correct data.

3.2.3. Deployment of Smart Contract. Smart contracts have decentered computations and storage based on blockchain. After the blockchain network is established, smart contracts will be deployed.

Step 1. After writing a smart contract, users use the browser compiler Remix to compile code into binary code.

Step 2. Users consume some tokens to deploy the compiled contract to the network, access to contract address of blockchain, and Application Binary Interface (ABI). ABI is a binary representation of the interface contract.

Step 3. When the user uploads the file, the IPFS address and Merkel Hash Tree [39] (MHT) root hash value will be obtained by preprocessing the file, and they are stored as a key-value pair in the data structure of the map by invoking smart contract by contract address and the ABI.

Step 4. When the user checks the file, users use the IPFS address of the file as the key to obtaining the MHT root hash value in the smart contract for comparison.

3.2.4. Destroy. Once deciding to delete VMA, tenants first call the kill function in smart contract for deleting the contract data and recovering the remaining tokens. And then, VMA starts the self-destruction of the module and the data will be rewritten overlay.

3.3. Blockchain Based Integrity Protection Mechanism. Based on the VMA architecture, the private chain is created and jointly safeguarded by the tenants in the edge cloud. The information can be traced back, tamper-resistant, and the trusted execution in the blockchain and smart contract. Therefore, this paper designs a data integrity monitoring program and blockchain integrity verification protocol based on the "challenge-response" model. The protocol is also based on the bilinear mapping of BLS [37] (Boneh-Lynn-Shacham) short signature verification [39, 40] and the mechanism is divided into three parts.

3.3.1. Pretreatment Stage. Get big primes p , $p \in Z_p$, set G_1, G_2 is Multiplication cycle group of prime number p, g_1 is the generator of G_1 , g_2 is the generator of G_2. There is a bilinear map, ℓ: $G_1 \times G_1 \longrightarrow G_2$. Randomly select $a, x \in Z_p$, $u = g_1^a$. The user generates a key pair $\{\mathrm{SK} = \{a, sk\}, \mathrm{PK} = \{g_1, u, pk\}\}$ locally, where the private key $sk = x$, public key pk: $v = g_2^x$.

Step 1. The user sends a request to connect the corresponding virtual machine agent. The VMA receives the user's request and then verifies whether it is valid or not. If the request is valid, the VMA will agree to connect. If not, a connection refusal response will be returned.

Step 2. Users upload files to VMA and VMA initializes data files. Firstly, the data information F is Partitioned into block $F = \{m_1, \ldots, m_i, \ldots, m_n\}$. Secondly, each block is divided into a segment, that is $m_i = \{m_{i,1}, \ldots, m_{i,j}, \ldots, m_{i,k}\}$, $1 \le j \le k$. Finally, call the tag generation algorithm for each data block. Generate a digital signature as follows:

$$\sigma_i = \left(H\left(b_i \| t_i\right) \cdot \prod_{j=1}^{k} g_1^{\left(a_j\right) \cdot h\left(m_{ij}\right)} \right)^x = \left(H\left(b_i \| t_i\right) \cdot \prod_{j=1}^{k} u_j^{h\left(m_{ij}\right)} \right)^x. \quad (1)$$

where H and h are hash function. H: $\{0,1\}^* \longrightarrow G_1$, h: $\{0,1\}^* \longrightarrow Z_p$. $a_j \in Z_p, x \in Z_p$. The data segment number b_i, timestamp t_i, $1 \le i \le n$. $\Phi = \{(\sigma_i) | 1 \le i \le n\}$ is a data information file F tag set of data blocks, the tag is stored in the database of the virtual machine agent.

Step 3. VMA uploads data F to store in the IPFS cluster and returns the IPFS address $F\_I\ d$, $F\_I\ d$ is unique identifier of data.

3.3.2. Data Integrity Monitoring Stage. After preprocessing of file integrity verification, VMA stores the digital signature of the data block in the database and computes the MHT root hash value according to the digital signature. The root value is deployed to the blockchain by invoking smart contracts. MHT is a kind of binary tree, as shown in Figure 3 [39]. The data tag value is stored only at the leaf nodes. The nonleaf nodes are obtained by the hash operation after linking the values of the left and right subnodes. Finally, the root hash value represents the integrity of the whole file.

Through the MHT root node, the tampering of any data block is detected to ensure the integrity of the file without the participation of other nodes of MHT. Meanwhile, MHT has only been a directed branch from the measured node to the MHT root node path, which can confirm whether the node exists in the data block or not, for example, verifying whether

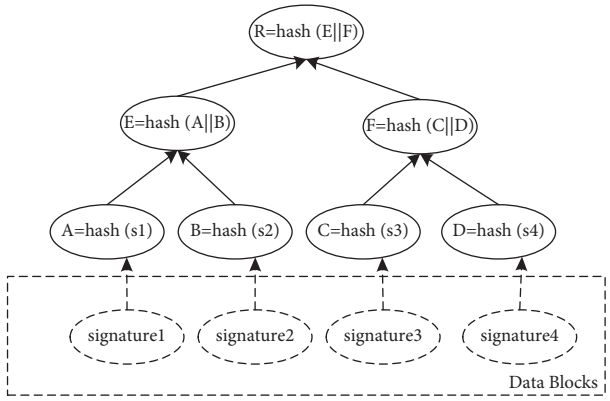

Figure 3: MHT structure diagram.

D is in the block according to the nodes C, E, and R. There are N data blocks in the MHT; hash computing is 2 $\log_2 N$; it can verify whether the data block has been tampered with or not.

The process of implementing the integrity monitoring mechanism is as shown in Figure 4. The process is as follows: The user uploads the file to the VMA for preprocessing. On the one hand, the file is divided into blocks, the tag is stored in the database, and the data tag of the file block generates the MHT; On the other hand, storing the file in the IPFS cluster gets the address based on context. By invoking the smart contract to save key-value pairs, the blockchain will monitor the value whether the file is modified.

3.3.3. Edge Cloud Data Integrity Verification Stage. When users are concerned that the data has been tampered with, users only need to challenge the ECSP. According to the ECSP's response, users can know whether the data is complete.

Step 1. The user sends a request of data integrity verification for the file to be detected. The request includes the data block set IDX = $\{i\ dx_i | 1 \le i \le c, c \le n\}$ and the corresponding random number set $R = \{r_i | i \in \text{IDX}, r \in Z_p\}$.

Step 2. Firstly, according to the challenge request, the VMA node queries the IPFS cluster for the IPFS unique flag $F\_I\ d$. Secondly, the VMA node creates a MA to migrate to the storage node to obtain the corresponding evidence of the data block. Variable c represents the total challenge number of data blocks to be detected, n is the total number of data blocks in the data block set.

Step 3. Storage node obtains the corresponding data block $\sum_{i \in \text{IDX}} h(m_{ij})$ by executing the MA task, returns the value to the VMA node, and calculates the total data block:

$$M = \sum_{j=0}^{k} \sum_{i \in \text{IDX}} h(m_{ij}). \tag{2}$$

According to the VMA node stored u, VMA calculates the total digital signature of the data block:

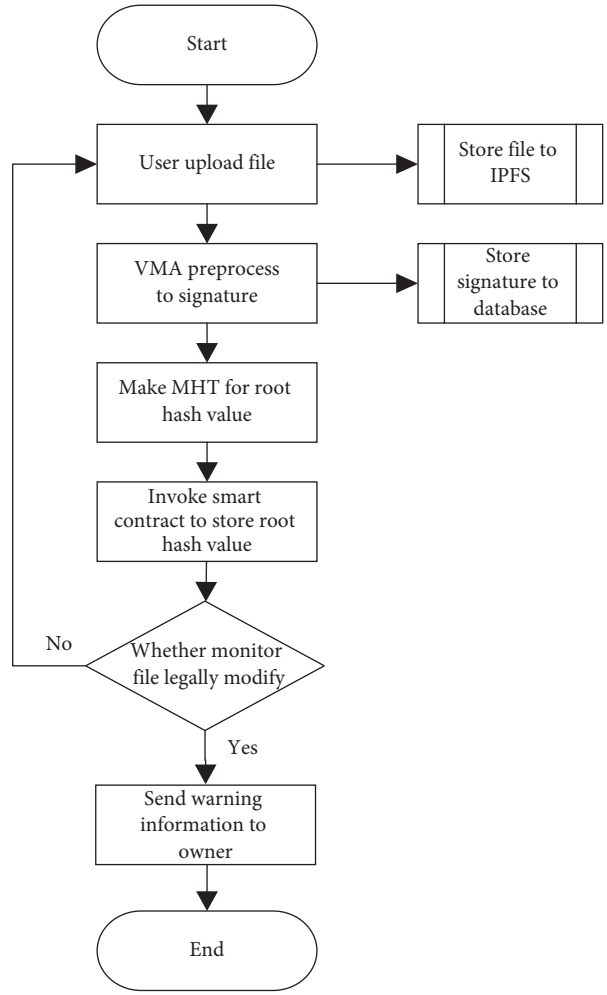

Figure 4: Integrity monitoring flow chart.

$$D = \prod_{j=0}^{k} u_j^{h(m)} = \prod_{j=0}^{j} u_j^{\sum_{j=0}^{k} \sum_{i \in \text{IDX}} h(m_{ij})}. \tag{3}$$

Step 4VMA reads the challenge data block tag value from its own database. And it then calculates the hash value of the corresponding challenge block number:

$$\begin{aligned} T &= \prod_{i \in \text{IDX}} \sigma_i^{r_i}, \\ B &= \prod_{i \in \text{IDX}} H(b_i \| t_i)^{r_i}. \end{aligned} \tag{4}$$

It generates evidence proof = $\{D, B, T\}$ and calculates:

$$l(B, v) \cdot l(D, v) \triangleq l(T, g_2). \tag{5}$$

If (5) holds, the supporting documents are complete.

Step 5. The user will receive the verification result of VMA and get the file MHT root hash value. If both values are equal, the verification result is credible.

The integrity verification stage is shown in Figure 5: some data blocks are randomly extracted by users. Users send a challenge to ECSP by VMA node. Firstly, the IPFS

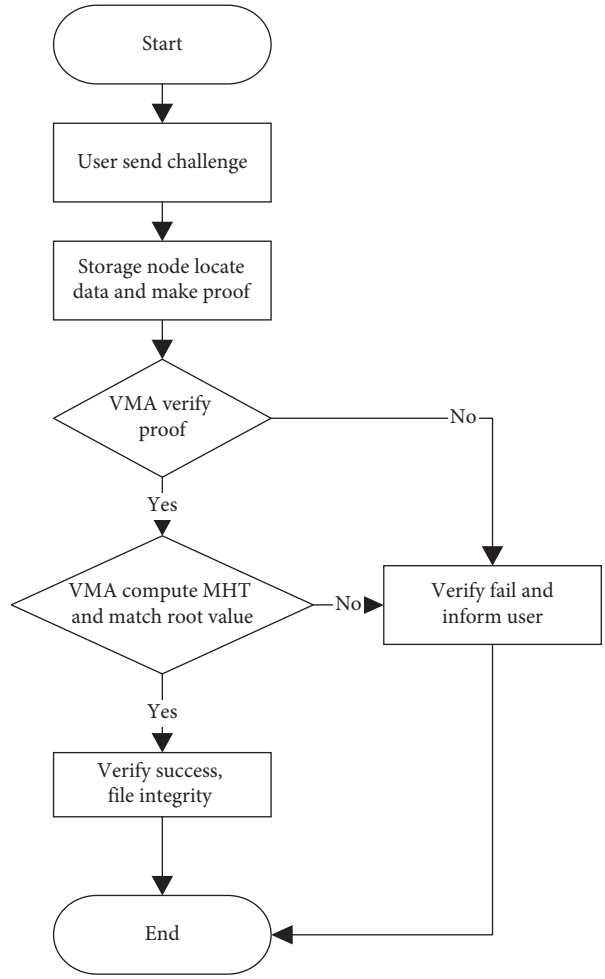

Figure 5: Integrity verification flow chart.

cluster fixes positions according to the challenge data block and generates the evidence back to the VMA. VMA verifies (5) and calculates the validity of the evidence. If valid, the second step validation will be performed to calculate whether the challenge block exists and whether the root hash value is consistent through the MHT. If consistent, the document is integrity, or the file is damaged.

4. Safety Certification

4.1. Analysis of Blockchain Integrity Monitoring Scheme. Based on the blockchain edge cloud data integrity mechanism, the following is considered for the normal modification and illegal tampering of validation results.

4.1.1. Attack Mode Analysis. ECSP attacks are divided into three levels.

First-level attacker: The attacker can break the security protection of the virtual machine agent and obtain the access control rights of the user, such as unauthorized users.

Second-level attacker: The attacker not only controls the virtual machine agent but also obtains the user's blockchain account address and key. The attacker invokes the intelligent contract interface and modifies the MHT hash value saved in the blockchain.

Third-level attacker: The attack value is illegally invaded into the IPFS cluster of the distributed storage system to tamper with and destroy the data. For example, administrators have the highest authority on data management, and if they are curious about user's data, they have direct access to the user's data at the storage node.

4.1.2. For Integrity Monitoring Mechanism Analysis of Normal and Illegal Tampering Validation Results. Normal modification: The user node sends an access request. Firstly, the VMA reads the data from the database and obtains the hash fingerprint of the corresponding data. Secondly, VMA obtains the data from the IPFS cluster and transmits the data to the user node. After the user modifies the data and invokes the smart contract, the VMA collects the affected data and generates a new digital label. Finally, VMA calls the smart contract interface to save the new MHT root hash to generate a new transaction. IPFS will update the database information.

Illegal tampering: As shown in Figure 6, if a third-level attacker attacks the storage data in the IPFS cluster, this

method can only select the data block number randomly from the sample integrity verification by the user. From the corresponding hash value found in the database, the file block is obtained by using IPFS, and the integrity verification operation is performed by the virtual machine agent.

If attacked by a first-level attacker, the data is illegally tampered with and the hash value and the digital signature corresponding to the data block in the database of the VMA change, so the root hash value generated by the MHT is also changed. Different from the value saved by the smart contract in the blockchain, the tampering failed.

If attacked by a second-level attacker, the attack value not only controls the virtual machine agent but also obtains the user's blockchain account and key and attempts to invoke the smart contract interface to modify the MHT root hash of the file. If successful, the transaction record will be left and saved by the other tenants; if it fails, the user will be warned through the smart contract.

4.2. Certificate of Integrity Agreement. This section will analyze the security of the scheme and propose that the system model may be attacked by three kinds of attacks [41] to solve the possible threats.

Theorem 1. *Proof of equality is whether it is established; if established, the document is complete; otherwise, the document has been tampered with.*

The proof is given as follows:

$$\begin{aligned}
&\ell(B,v)\cdot\ell(D,v)\\
&=\ell\left(\prod_{i\in\mathrm{IDX}}H\left(b_i\|t_i\right)^{r_i},v\right)\cdot\ell\left(\prod_{j=0}^{k}u^{\sum i=0^{k}}\sum_{j\in\mathrm{IDX}}h\left(m_{ij}\right),v\right),\\
&=\ell\left(\prod_{i\in\mathrm{IDX}}H\left(b_i\|t_i\right)^{r_i},v\right)\cdot\ell\left(u^{\sum_{i\in\mathrm{IDX}}h(m_i)},v\right),\\
&=\ell\left(\prod_{i\in\mathrm{IDX}}H\left(b_i\|t_i\right)^{r_i}\cdot u^{\sum_{i\in\mathrm{IDX}}h(m_i)},v\right),\\
&=\ell\left(\prod_{i\in\mathrm{IDX}}\left(H\left(b_i\|t_i\right)\cdot u^{h(m_i)}\right)^{r_i},g_2^{x}\right),\\
&=\ell\left(\prod_{i\in\mathrm{IDX}}\left(H\left(b_i\|t_i\right)\cdot u^{h(m_i)}\right)^{xr_i},g_2\right),\\
&=\ell\left(\prod_{i\in\mathrm{IDX}}t_i^{r_i},g_2\right),\\
&=\ell(T,g_2).
\end{aligned}\tag{6}$$

Theorem 2. *Forge attacks. If the data owner reuses a certain secret value for different versions of data when generating a signature, then in the storage node, the ECSP may forge the data signature of the data block to deceive the verifier.*

Proof. In the integrity verification mechanism, ECSP is not feasible to forge audit evidence in order to pass verification.

Game Definition: The user sends a challenge message to the storage node of the ECSP through the VMA:

$$\mathrm{chal}=\left(\mathrm{IDX}=\left\{i\ dx_i|1\le i\le c,c\le n\right\},R=\left\{r_i|i\in\mathrm{IDX},r\in Z_p\right\}\right).\tag{7}$$

In order to verify (5), the ECSP should send based on the correct file, Audit evidence, but the ECSP constructed the evidence from the wrong data.

$$\begin{aligned}
&\mathrm{proof}=\left\{D\prime,B,T\right\},\\
&D=\prod_{j=0}^{k}u_j^{h(M)}=\prod_{j=0}^{k}u_j^{\sum}j=0^{k}\sum_{i\in\mathrm{IDX}}h\left(m_{ij}^{\prime}\right),\\
&M^{\prime}=\sum_{j=0}^{k}\sum_{i\in\mathrm{IDX}}h\left(m_{ij}^{\prime}\right).
\end{aligned}\tag{8}$$

□

Definition $h(\nabla m_i)=h(m_i^{\prime})-h(m_i)$, $i\in\mathrm{IDX}$, at least one element is nonzero. If ECSP falsified evidence still passes VMA verification, ECSP wins the games, Otherwise, it fails.

Suppose ECSP won the game, according to the verification (5),

$$\ell(B,v)\cdot\ell\left(D^{\prime},v\right)\triangleq\ell\left(T,g_2\right).\tag{9}$$

According to the dual mapping $u^{\sum_{i\in\mathrm{IDX}}h(m_i)}=u^{\sum_{i\in\mathrm{IDX}}h(m_i^{\prime})}\Rightarrow u^{\sum_{i\in\mathrm{IDX}}h(\nabla m_i)}=1$, G is a step for the multiplicative cyclic group of a prime number p. There are elements $c_1,c_2\in G,\exists x\in\mathbb{Z}_p,c_2=c_1^{x}$. Then $u=c_1^{\alpha}c_2^{\beta}\in G$, $u^{\sum_{i\in\mathrm{IDX}}h(\Delta m_i)}=c_1^{\alpha\sum_{i\in\mathrm{IDX}}h(\Delta m_i)}\cdot c_2^{\beta\sum_{i\in\mathrm{IDX}}h(\Delta m_i)}=1$, $\sum_{i\in\mathrm{IDE}}h(\Delta m_i)\neq 0,\beta\neq 0,x=-\alpha/\beta$. The probability of $\beta=0$ is $1/p$; then for the DL assumption $1-1/p$, the probability of solving contradicts the DL conjecture. Therefore, ECSP is proved as unforgeability.

Theorem 3. *Alternative attack, when the data block m_i or signature t_i is lost, the ECSP may replace the user's challenge with another valid data and data signature.*

Game Definition: The user sends a challenge message to the storage node of the ECSP through the VMA:

$$\begin{aligned}
\mathrm{chal}=&\left(\mathrm{IDX}=\left\{i\ dx_i|1\le i\le c,c\le n\right\},\right.\\
&\left.R=\left\{r_i|i\in\mathrm{IDX},r\in Z_p\right\}\right).
\end{aligned}\tag{10}$$

In order to pass the verification equation above, the ECSP should send audit evidence $\mathrm{proof}=\left\{D\prime,B,T\right\}$ based on the correct document F. The ECSP constructs data block evidence

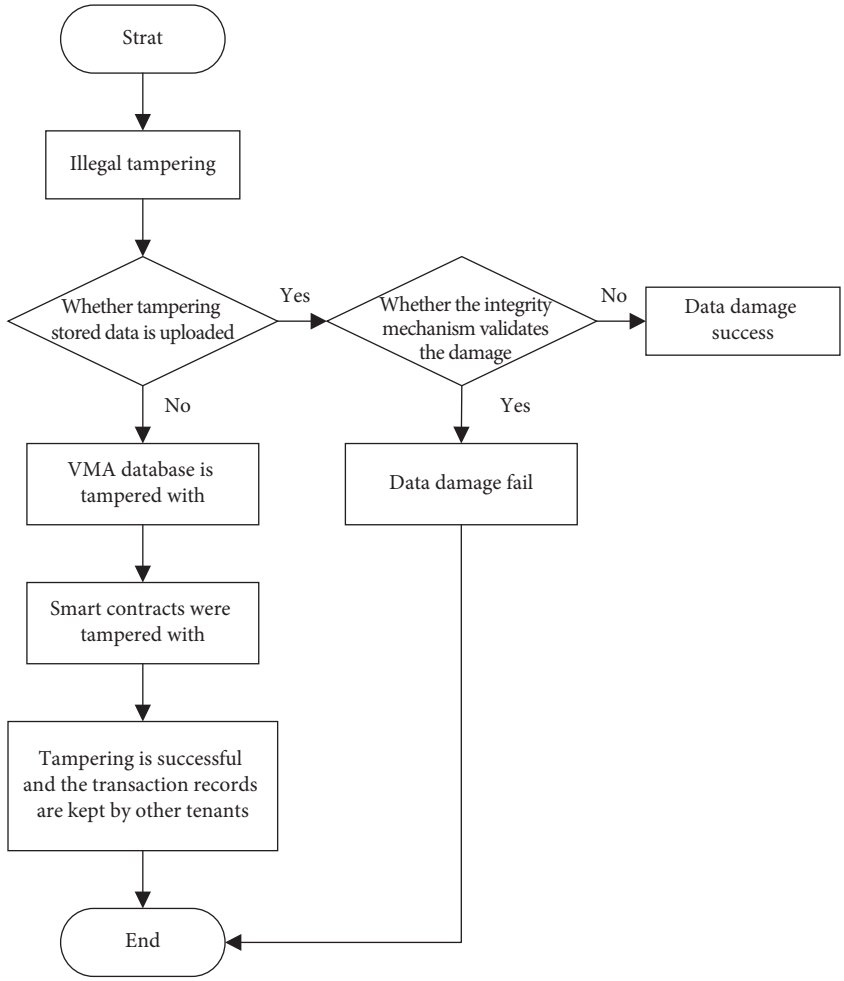

Figure 6: Illegal tampering with the flow chart.

$f-th(f \in IDX)$ instead of $i-th(i \in IDX)$. If it is proved by VMA and can still be verified, ECSP will win the game; otherwise, it will fail. According to the properties of the bilinear map pair,

$$\begin{aligned}\ell(B,v)\cdot\ell(D\prime,v) &= \ell\left(\prod_{i\in \mathrm{IDX}} H\left(b_i\|t_i\right)^{r_i},v\right)\cdot\ell\left(\prod_{j=0}^{k} u_j^{\sum} j=0^{k}\left(\sum_{i\in \mathrm{IDX}\&i\neq f} h\left(m_{ij}\right)+h\left(m_{fj}\right)\right),v\right),\\ &= \ell\left(\prod_{i\in I\ DX} H\left(b_i\|t_i\right)^{r_i},v\right)\cdot\ell\left(u^{\sum_{i\in \mathrm{IDX}\&i\neq f}^{h(m_i)}}\cdot u^{h\left(m_f\right)},v\right),\\ &= \ell\left(\prod_{i\in \mathrm{IDX}\&i\neq f}\left(H\left(b_i\|t_i\right)\cdot u^{h\left(m_i\right)}\right)^{xr_i}\cdot\left(H\left(b_f\|t_f\right)\cdot u^{h\left(m_f\right)}\right)^{xr_f},g_2\right).\end{aligned} \tag{11}$$

If the above formula is established, b_i is on behalf of the data section number, then $b_i = b_f$, $t_i = t_f$. Because the definition $f \neq i$, then $t_i \neq t_f$. Therefore $H(b_i\|t_i) \neq H(b_f\|t_f)$. ECSP replaces data signature failure.

Theorem 4. *Replay attacks; ECSP may not need to retrieve stored data; use the previous response to the evidence or other information to generate this evidence.*

Proof. The repeated attacks are defined as follows. The VMA sends a challenge request to the ECSP.

$$\mathrm{chal} = \left(\mathrm{IDX} = \{i\ dx_i | 1 \le i \le c, c \le n\}, R = \left\{r_i | i \in \mathrm{IDX}, r \in Z_p\right\}\right). \tag{12}$$

ECSP responds with an audit certificate proof $= \{D', B, T\}$. In the process of generating a proof, each data block $j-th(j \in \mathrm{IDX})$ is replaced by the previous

information. This paper uses single quotes to separate the previous parameter from the correct parameter, for example: m_j' is the previous data block, m_j is the correct data block. If this proof is still validated by third-party audits, ECSP will resist replay attacks.

This proof is similar to Theorem 3, the same data block timestamps cannot be consistent. That is $H(b_j \| t_j) \neq H(b_j \| t_j')$, ECSP will fail. □

5. Prototype System

This paper uses advanced language (Solidity), which is designed to compile code that generates code that can run on the blockchain. The entire system is divided into three parts: Web client, VMA server, and blockchain API. As shown in Figure 7, a Web client mainly allows users to upload files, generate accounts address of blockchain and initiate challenges integrity verification operations. The VMA server can mainly preprocess files, respond to the challenge of integrity verification, establish MHT, and interact with the blockchain network by the blockchain API, such as account address generation, smart contract creation, and IPFS storage.

5.1. System Overall Process. Figure 8 shows three important functions of the prototype system: interacting with the account address generation, completing pretreatment, and verifying data integrity.

5.2. Function to Achieve

5.2.1. Web Client Implementation. Upload files: Upload files to the edge cloud VMA server, set the conditions of file division, and control the size of data blocks.

Download File: Save the file to the edge cloud IPFS cluster and obtain the source file based on the IPFS file address.

Initiate the challenge: The user selects the appropriate number sent to the VMA for integrity challenges according to the total number of files.

Register account address of blockchain: Provide user name and password to be completed by the VMA server registration.

5.2.2. VMA Server Implementation. Create a smart contract: Use the blockchain account address call ABI of the smart contract, and spend a certain token to generate a new contract address which is used to save the MHT root hash.

Register accounts address of blockchain: The account of each blockchain is composed of a pair of public and private keys, and the account address is 20-byte public key derived. The account uses public key encryption to sign the deal in order to send a secure authentication identity of the person in the blockchain network. The private key is encrypted with the password provided by the user. All blockchain operations are based on the address, and the same user can register multiple account addresses to prevent privacy disclosure.

Query Information on the blockchain network: According to the Transaction id or block number, the user can query information of the blockchain to track changes to files and postaudit.

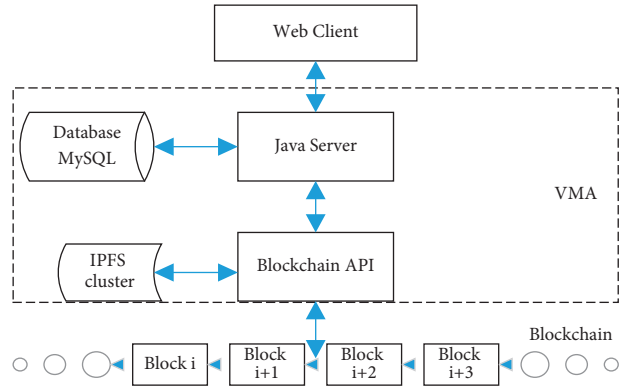

Figure 7: System frame diagram.

Preprocess file: The file is divided into data blocks according to user requirements and then generates a digital signature to calculate the MHT root hash value.

Verify challenge: According to the number of Web client challenges, VMA obtains the source data block from the ECSP, calculates the evidence, and verifies the data integrity.

5.2.3. Database Implementation. VMA database includes three tables, namely: (1) Fileinfo table, and the digital data signatures are being uploaded for integrity verification and MHT generation; (2) Public table, record public information and selected random number; (3) Users table, record the account address and transaction id and other related information.

6. Results and Discussion

6.1. Experimental Setup. The following contents will design experiments on this mechanism named Blockchain Proof of Data Possession (BPDP). Four virtual machines are used to simulate the VMA to form a blockchain network. Each virtual machine has the whole module for integrity verification. Users interact with the VMA through the web system. The integrity verification module uses JPBC (Java Pairing Based Cryptography) version 2.0.0. The elliptic curve uses MNT d159 curve. The basic domain size is 159, and the embedding degree is 6. The safety parameter selected experiment is 80 bit. The experiment in the system randomly generated a fixed size of the file F, and each experimental result takes the experimental average of 30 times.

6.2. Performance Analysis. At first, the integrity verification protocol is performed based on the accuracy of sample analysis. Assuming the total number of data blocks in the edge server is n if the number of error data blocks is e and corrupted data block ratio is $p_b = e/n$. Assuming that t is the ratio of the number of data blocks in each challenge to the total number n, then the probability of illegal tampering detected each time is as follows:

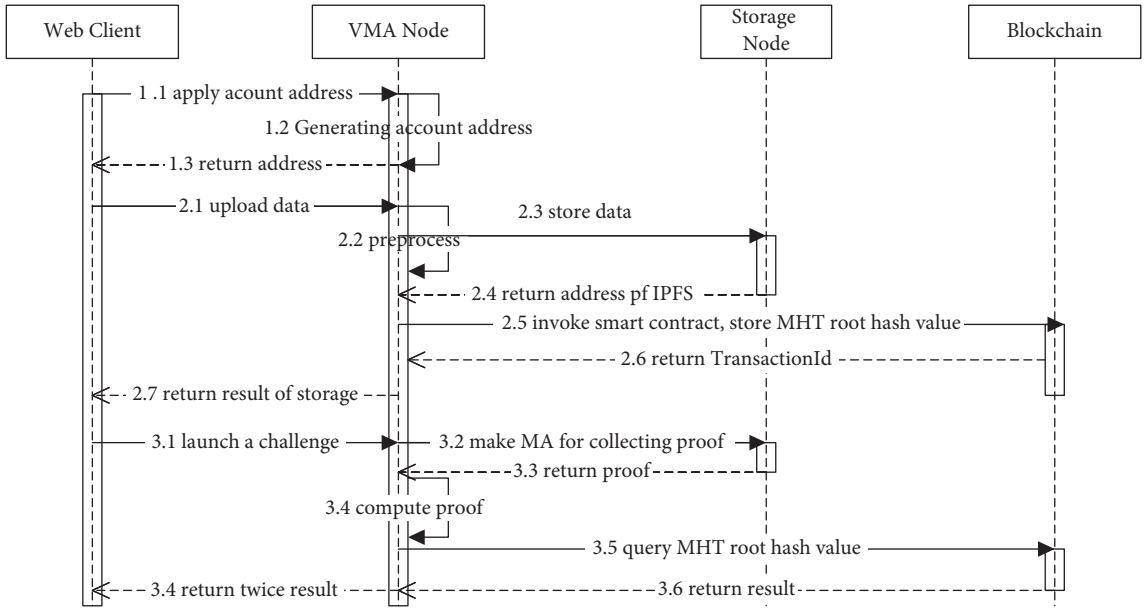

Figure 8: Prototype system time sequence diagram.

$$\begin{aligned} P &= P\{X \geq 1\} \\ &= P\{X = 0\} \\ &= 1 - \frac{n-e}{n} \cdot \frac{n-1-e}{n-1} \cdot \cdots \cdot \frac{n-t \cdot n-e}{n-t \cdot n} \geq 1 - \left(\frac{n-e}{n}\right)^{t \cdot n} \\ &= 1 - (1-p_b)^{t \cdot n}. \end{aligned} \tag{13}$$

As shown in Figure 9, if the number of error data blocks with the total number of data blocks ratio is 0.1%, the accuracy of 99%, and the total number of data blocks is 10,000, the number of challenge blocks is 4600. As shown in Figure 10, if the ratio of damage is 1%, then the number of challenge blocks is 460. Therefore all integrity protocols perform relatively poorly with less damage ratio. In the paper preprocessing, MHT is constructed to store the root hash of the file into the blockchain, which is twice used to ensure the file is not tampered with after sample integrity verification.

The security parameter selected in this paper is 80 bit, meaning $|p| = 160$. The storage cost of data signature is $n * p/8$, n is the number of data blocks. In order to achieve data dynamic operation, the establishment of an index hash table (IHTCost) spends storage cost is $n * (2 * p + 2 \log n)/8$. When the number of data segments is fixed, the larger the data segment and the smaller the number of formed data blocks will reduce the storage costs of the index hash table, as shown in Figure 11.

The most critical module of the prototype system is the edge cloud data integrity verification module. As shown in Figure 12, when the file block is too small, resulting in a dramatic increase in the number of file blocks and consumes longer pretreatment time. If the block is large, the number of

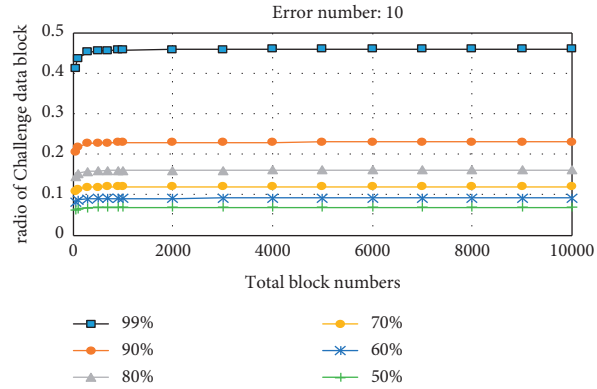

Figure 9: Challenge data block scale when the number of error block is 10.

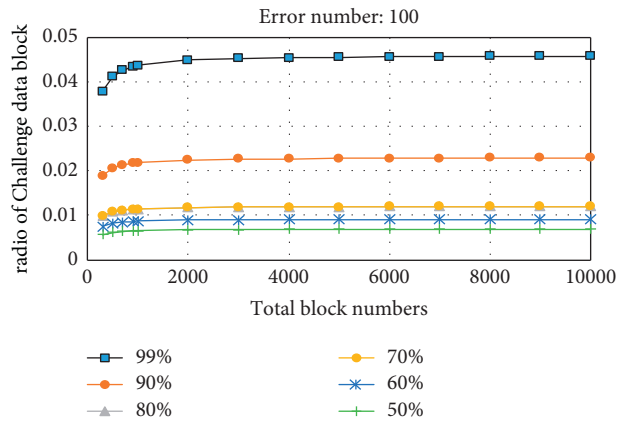

Figure 10: Challenge data block scale when the number of error block is 100.

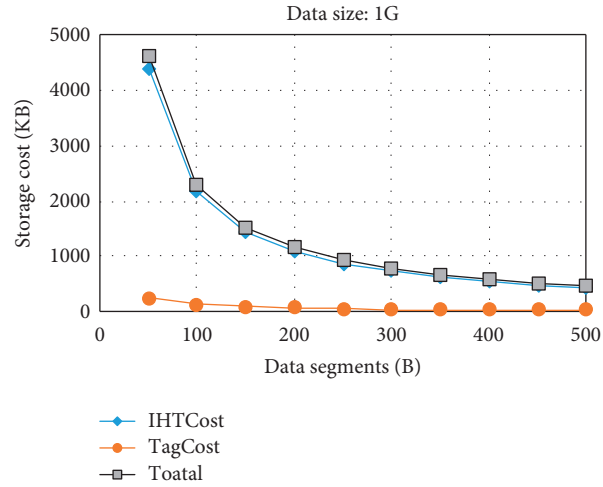

Figure 11: Challenge data block scale when the number of error blocks is 100.

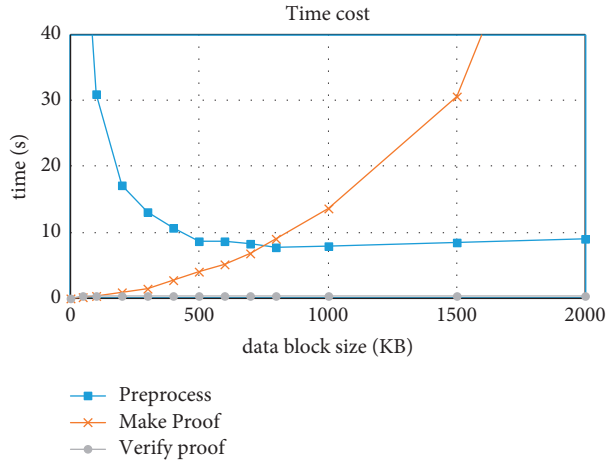

Figure 12: Prototype system time costs.

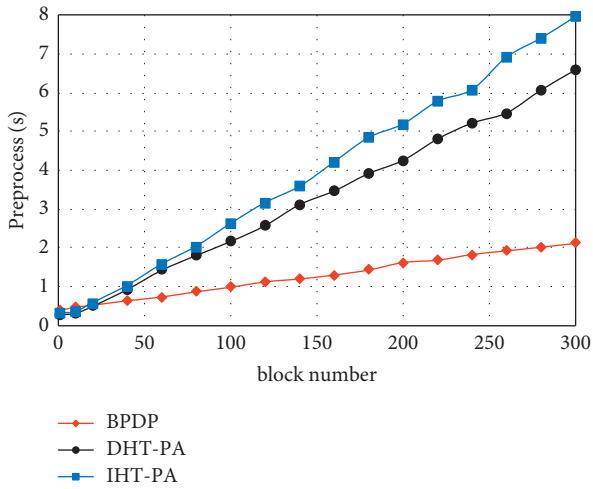

Figure 13: Verification time comparison.

data segments increases dramatically when the data block is divided into data segments resulting in an increase in the time for the generation of evidence. The file is set to 1G, and 12S completes a series of integrity validation when the number of data blocks is set reasonably.

Next, this paper analyzes the time costs of performing an integrity verification in DHT-PA [10], IHT-PA [42], and the BPDP. According to the accurate analysis of the verification, it is assumed the case when the error ratio is 1%. This paper selects the appropriate number of challenge blocks in order to achieve 99% accuracy. As shown in Figure 13, the experiment shows that the preprocessing time is proportional to the number of data blocks when processing the same size data block (50 KB). The result analyzes that the time costs of this paper are better than that of the same data block.

7. Conclusions

The above analysis shows that users store data on the edge cloud server and delegate the integrity verification of the remote data to the VMA so as to reduce the burden on users and eliminate the potential threats of third-party auditors. VMA itself is in the edge cloud and reaches a protocol consensus through information exchange in an unreliable, potentially threatening network, enabling trusted integrity

verification in an untrusted environment, protecting user data integrity, and preventing data from being illegitimate tampered with. In addition, the blockchain can save the interaction information of user and ECSP and record the nonrepudiation information which is manipulated by users' operations in the edge cloud environment so as to collect effective, reliable legal evidence to establish a perfect accountability mechanism. The next step will be to implement the access control of smart contracts according to the scheme, set access rights, and improve the control of user data, so as to better protect user data. It is hoped that the scheme can finally be put into production.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62072255 and the Postgraduate Research and Practice Innovation Program of Jiangsu Province.

References

[1] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] M. Du, Y. Wang, and K. Ye, "Algorithmics of cost-driven computation offloading in the edge-cloud environment," *IEEE Transactions on Computers*, vol. 69, no. 10, pp. 1519–1532, 2020.

[3] K. Gai, J. Guo, and L. Zhu, "Blockchain meets cloud computing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.

[4] W. Shi, J. Cao, and Q. Zhang, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[5] E. El Haber, T. M. Nguyen, and C. Assi, "Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3407–3421, 2019.

[6] C. Li, J. Bai, and Y. Chen, "Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system," *Information Sciences*, vol. 516, pp. 33–55, 2020.

[7] Y. Fan, *Research on Cloud Data Integrity Verification and Data Recovery*, Chongqing University, Chongqing, China, 2016.

[8] S. Yan, Y. Chen, and P. Liu, "Security protection mechanism of virtual machine computing environment under the cloud computing," *Journal on Communications*, vol. 36, no. 11, pp. 102–107, 2015.

[9] C. C. Erway, A. Küpçü, C. Papamanthou, and T. Roberto, "Dynamic provable data possession," *ACM Transactions on Information and System Security*, vol. 17, no. 4, p. 15, 2015.

[10] H. Tian, Y. Chen, and C. Chang, "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 701–714, 2017.

[11] G. Xu, Y. Bai, C. Yan, and Y. Yang, "Check algorithm of data integrity verification results in big data storage," *Journal of Computer Research and Development*, vol. 54, no. 11, pp. 2487–2496, 2017.

[12] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu, and H. Thaier, "Preserving balance between privacy and data integrity in edge-assisted Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2679–2689, 2019.

[13] D. W. Chadwick, W. Fan, G. D. D. Costantino et al., "A cloud-edge based data security architecture for sharing and analysing cyber threat information," *Future Generation Computer Systems*, vol. 102, pp. 710–722, 2020.

[14] X. Li, S. Liu, F. Wu, and P. C. R. Joel, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.

[15] T. Wang, Y. Mei, and W. Jia, "Edge-based differential privacy computing for sensor-cloud systems," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 75–85, 2020.

[16] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, https://bitcoin.org/bitcoin.pdf.

[17] Ethereum, "Ethereum whitepaper," 2020, https://ethereum.org/en/whitepaper.

[18] Hyperledger, "An introduction to hyperledger," 2018, https://www.hyperledger.org/wpcontent/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf.

[19] W. Tsai, L. Yu, and R. Wang, "Blockchain application development techniques," *Journal of Software*, vol. 28, no. 6, pp. 1474–1487, 2017.

[20] A. Kosba, A. Miller, E. Shi, Z. Wen, and P. Charalampos, "Hawk:the blockchain model of cryptography and privacy-preserving smart contracts," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 839–858, IEEE, San Jose, USA, May 2016.

[21] Ibm, "Research leading block chain use cases," https://www.ibm.com/blockchain/use-cases/.

[22] Microsoft, "What are blockchain-enabled digital ecosystems," 2020, https://azure.microsoft.com/en-us/resources/what-are-blockchain-enabled-digital-ecosystems/.

[23] Amazon, "Amazon managed blockchain," https://amazonaws-china.com/cn/managed-blockchain/.

[24] Y. Tian, J. Yuan, and H. Song, "Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones," *Journal of Information Security and Applications*, vol. 48, Article ID 102354, 2019.

[25] T. Wang, P. Wang, S. Cai, Y. Ma, A. Liu, and M. Xie, "A unified trustworthy environment establishment based on edge computing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6083–6091, 2019.

[26] D. Yue, R. Li, and Y. Zhang, "Blockchain-based verification framework for data integrity in edge-cloud storage," *Journal of Parallel and Distributed Computing*, vol. 146, pp. 1–14, 2020.

[27] E. Bonnah and J. Shiguang, "DecChain: a decentralized security approach in Edge Computing based on Blockchain," *Future Generation Computer Systems*, vol. 113, pp. 363–379, 2020.

[28] Z. Ma, X. Wang, D. K. Jain, H. Khan, H. Gao, and Z. Wang, "A blockchain-based trusted data management scheme in edge

computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2013–2021, 2020.

[29] J. Kang, R. Yu, X. Huang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.

[30] K. Gai, Y. Wu, and L. Zhu, "Differential privacy-based blockchain for industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4156–4165, 2020.

[31] B. Li, Q. He, F. Chen, H. Jin, X. Yang, and Y. Yang, "Auditing cache data integrity in the edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1210–1223, 2020.

[32] W. Tong, B. Jiang, F. Xu, X. Lu, and Z. Sheng, "Privacy-preserving data integrity verification in mobile edge computing," in *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1007–1018, IEEE, Dallas, TX, USA, July 2019.

[33] G.-X. Liu, L.-F. Shi, and D.-J. Xin, "Data integrity monitoring method of digital sensors for Internet-of-Things applications," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4575–4584, 2020.

[34] G. S. Aujla, A. Singh, and M. Singh, "BloCkEd: blockchain-based secure data processing framework in edge envisioned V2X environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5850–5863, 2020.

[35] J. Wu, *The Research of Fault Detection and Event Detection for Wireless Networks*, Nanjing University of Posts and Telecommunications, Nanjing, China, 2013.

[36] X. Xu, P. Gong, Y. Zhang, and C. G. Bi, "Mobile-agent-based composite data destruction mechanism for cloud-P2P," *Computer Science*, vol. 42, no. 10, pp. 138–146, 2015.

[37] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[38] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *Eprint Arxiv*, vol. 07, no. 2, pp. 23–34, 2014.

[39] S. Tan, Y. Jia, and W. Han, "Research and development of provable data integrity in cloud storage," *Chinese Journal of Computers*, vol. 38, no. 1, pp. 164–177, 2015.

[40] C. Liu, J. Chen, L. T. Yang et al., "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2234–2244, 2014.

[41] Z. Qin, S. Wu, and H. Xiong, "A review on data integrity protocols for data storage in cloud computing," *Netinfo Security*, vol. 7, pp. 1–6, 2014.

[42] S. S. H. H. J. Yau, "Dynamic audit services for outsourced storages in clouds," *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227–238, 2013.