Distributed Electronic Data Storage and Proof System Based on Blockchain

Jitao Wang, Guozi Sun(✉), Yu Gu, and Kun Liu

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China
joten_wang@qq.com, sun@njupt.edu.cn, 243477384@qq.com, liukun_it@163.com

Abstract. In the context of the Internet, whether it is daily business or social networking, the penetration of electronic data is ubiquitous. Internet companies, financial institutions, government agencies and many other fields, more and more documents, notices, contracts, transaction vouchers, technology and trade secrets are stored in the form of electronic data. However, the existing traditional electronic data storage and proof systems are often encountered with third-party trust crisis and potential data security risks. To cope with these challenges, a distributed electronic data storage and proof system is designed, making use of the core features of the blockchain's decentralization and non-tampering to effectively solve the tampering and security problems of electronic data storage and proof. The system encodes and fragments information using Reed-Solomon code. And this system provides users with data uploading, downloading, querying, comparing and authorizing services. By using the system interaction, smart contracts are compiled to anchor key data information on the main chain, ensuring the non-tampering of electronic data. In the meantime, the access rights of different users to electronic data are restricted accordingly. Finally, based on an improved RFM model, the distributed storage nodes are determined to achieve load balancing of storage nodes. It also increases the high availability of the system.

Keywords: Blockchain · Smart contract · Decentralization · Distributed storage · Electronic data · Load balancing

1 Introduction

In the past few years, blockchain technology has gained tremendous growth, mainly attributed to the success of Bitcoin cryptocurrency. A blockchain (also known as a distributed ledger) is essentially an additive database maintained by a set of nodes that are not fully trusted by each other. Since the blockchain is kept running in the decentralized network, it provides a constant source of power for the transaction, verification, and interconnection of the blockchain. However, with the continuous development of application scenarios, the design

X. Si et al. (Eds.): CBCC 2019, CCIS 1176, pp. 48–67, 2020.
https://doi.org/10.1007/978-981-15-3278-8_4

of Bitcoin has the problems with lack of Turing complete, lack of account preservation, excessive resource consumption and limited efficiency. Bitcoin has not been applied in many blockchain scenarios, so in this case, a multi-layered, cryptographic-based open source technology agreement Ethereum comes into being. It integrates different functional modules through the overall design and is a comprehensive platform for creating and decentralized applications [1–3].

Electronic data is the product of modern technology, which requires us to store it and prevent from tampering. Electronic data storage and proof is a system structure that is easy to browse, easy to prove, easy to identify, and easy to save. First, it stores the data well according to the data type and has a good guarantee for the credibility and integrity of the data [4]. The storage and proof of electronic data is convenient for storing and verifying, which also provides effective data sharing securely. Blockchain technology can provide a complete security encryption technology and user authentication system [5,6].

At present, there are several problems in the electronic deposit certificate that need to be solve [7,8]:

(1) The degree of automation in the process of depositing certificates is not high.
(2) The risk of electronic data storage and proof is large.
(3) The legal processing procedure of third-party organizations is cumbersome.
(4) The security of electronic data is limited.
(5) There is lack of trust between the two organizations.

In this paper, we present a secure, scalable electronic data storage and proof system. We use data and user mapping to ensure efficient access control to the electronic data pool. We design a blockchain-based data storage and proof scheme that allows data users/owners to access electronic data from an electronic repository after authentication. The data storage mainly performs fragment redundancy algorithm and distributed storage to ensure data security, and the system introduces a user point mechanism to ensure system load balancing. The verification and subsequent services are enclosed within the system, written to the block, and become part of the blockchain.

2 Related Work

In this section, an overview of the systematic research related to blockchains is presented, with an emphasis on the application of blockchain technology. The application of blockchain technology in real life scenarios involves medical, insurance, copyright protection, and the Internet of Things.

Qi et al. briefly solve the access control management problem in medical data sharing system in their research. It mainly designed a blockchain-based data sharing scheme, allowing data users/owners to visit the electronic medical records from shared repositories after identity authentication and encryption key authentication. Sifah et al. propose a blockchain-based shared medical data solution, with a focus on providing data access control, source, and audit, and sharing medical data among cloud service providers [9].

There is also consideration in security, cloud storage and other aspects: Liang et al. propose a decentralized and trusted cloud data origin architecture using blockchain technology. Blockchain-based data sources can provide tamper-proof records, transparency of data in the cloud, and enhanced privacy and usability of source-based data [10]. An Binh Tran et al. propose a browser-based tool for managing and deploying user registrations and calling smart contracts on the blockchain.

In the application research of electronic data storage and proof based on blockchain, Li et al. study how to combine business with blockchain technology and propose a method to optimize current data storage from the application scenarios of electronic data storage to provide effective services for users. Li et al. explore the collection meaning of electronic evidence in cybercrime in the collection and preservation analysis of cybercrime, analyze the particularity of electronic evidence collection in cybercrime, and propose the method of collecting and preserving electronic evidence [11].

The form of electronic data is confusing, and the data format cannot be effectively unified, which brings extra work during data storage process. There is a big data security risk in the electronic data storage in that the centralized storage method may cause the data to be tampered with and lost, making the whole system not completely credible. Secondly, the electronic data has a long waiting time for obtaining the verification result, and the obtained result is sluggish, so that the user cannot obtain the result in time. The corresponding information cannot be given in time, and the system efficiency is limited [10, 12]. Therefore, this paper is devoted to solving various problems encountered in electronic data, and proposes a research based on blockchain technology to solve the problem of electronic data storage and proof. The main contributions of this paper are as follows:

(1) This paper proposes a system that combines electronic data storage with blockchain technology. The storage and proof of electronic data is used to store and verify various types of data. The blockchain technology is used to fix and save the acquired electronic data [13].
(2) This paper applies the distributed storage method of electronic data and performs redundant fragmentation on data to ensure the security of data storage. We introduce the credit system in the system, according to the user uploading the storage and proof and providing the storage method to integrate the data changes, maintaining the load balance of the system, ensuring the security and stability of the system [14].

3 Technology Architecture

The design of the system adopts the idea of “high cohesion and low coupling”. The main functions of the whole system have four layers. From top to bottom, it is the business layer, the logic layer, the intelligent contract layer, and the blockchain layer [15–17], as shown in Fig. 1.

(1) Application layer: The application layer mainly includes the front-end UI, the display layer and the business layer in Fig. 1, and the front-end UI provides a visualized web interface for the user access system. It receives the request submitted by the user, performs simple pre-processing, and sends the request to the logic layer for core calculation. After the calculation is completed, the data information is received from the logical layer, and is intuitively fed back to the user through the web interface [18,19]. The user can be a customer who needs to maintain data, or a third party that needs to download data for notarization.
(2) Logical layer: It is the implementation layer of the core functions of the system. According to the six interfaces provided by the application layer, the logic layer respectively gives the implementation method of the corresponding functional modules [20]. Among them, the TCP-based Socket multithreaded concurrent module is the framework foundation for the entire system to run smoothly. The system uses this module to achieve reliable transmission of data between different nodes. Based on the above basic framework, the system introduces the coding and decoding of Reed-Solomon codes, and node selection module used in the implementation of file uploading and downloading modules and introduces the user node performance test module to determine its advantages and disadvantages. It introduces the Hash comparison module to determine whether the file has been maliciously tampered with. Finally, it introduces user registration and point module to complete the management of user information.
(3) Smart contract layer: The smart contracts deployed on the Ethereum platform. As a bridge between the logical layer and the blockchain layer, the smart contract layer anchors the calculation results of the logical layer (such as the electronic data and its fragmentation fingerprint information, the user node's point information, etc.) to the blockchain layer storage area. In the process of writing smart contracts, the system defines several structures (such as File, Record, User, etc.) to store key information of electronic data in the form of customized data. This method significantly improves the efficiency of electronic data query and enhances the readability of electronic data [21,22].
(4) Blockchain layer: As a decentralized database of the system, the data information generated by the logic layer is stored. The network layer undertakes to verify the transaction information, generate new blocks, and maintain the stable operation of the blockchain network. The data layer stores all the key information uploaded by the entire system [23].

In this paper, based on the new idea of block chain, the distributed storage records of electronic data are placed on the block chain, and combined with redundant slice algorithm, time stamp, hash algorithm, Reed-Solomon code, fuzzy analytic hierarchy process, ideal base point, improved RFM model and smart contract, the distributed storage system based on block chain is set up and built.

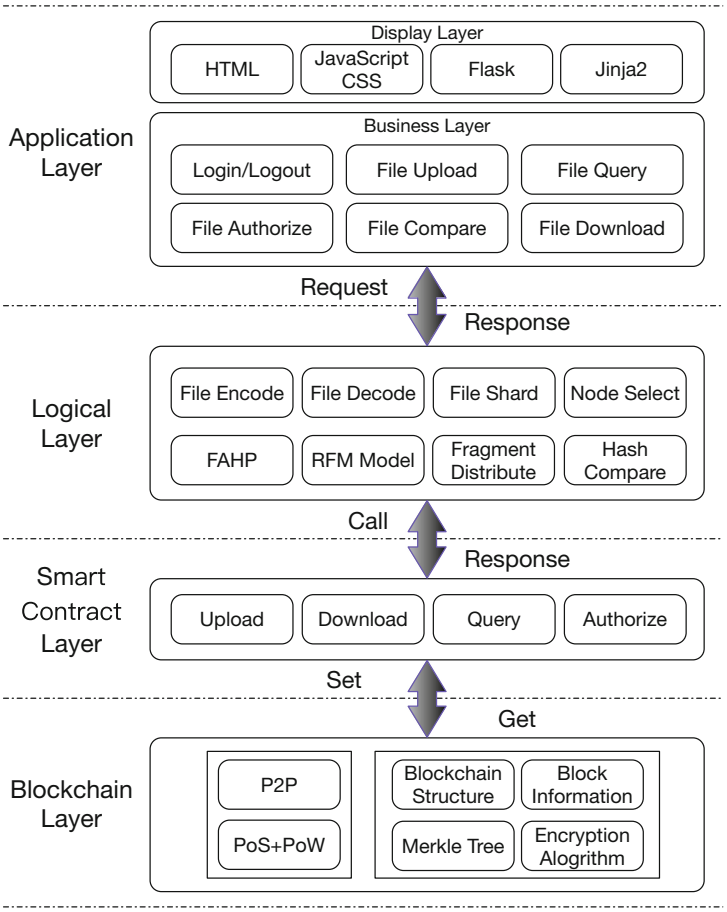

Fig. 1. System function architecture diagram

The system mainly adopts the data redundancy fragmentation technology, which divides electronic data into n information slices and M redundant pieces, and then carries out distributed storage of data slices to ensure the security of data information, collect the host server of the system, combine fuzzy analytic hierarchy process and ideal base points to test the network comprehensive performance parameters of the system storage host to ensure load balancing of system information storage.

Firstly, it uses the P2P network technology to implement the centralization of the system, which is based on the decentralization characters of smart contracts. Secondly, it uses the hash algorithm to ensure the integrity of the data. Finally, the consensus algorithm is used to ensure the consistency of the data between the nodes.

In this way, a decentralized and verifiable distributed storage system is designed. Based on the system, the smart contracts can automatically realize the transaction processing and saving mechanism of the electronic data storage and proof under the participation of the two or more parties, and the third party institutions such as the public security law [24–26].

3.1 Blockchain Technology

A blockchain is a transaction database shared by all nodes that participate in the network based on a transaction protocol. The blockchain contains every transaction that has been executed in the system. Based on this, people can find information about any address at any time. If the blockchain is used as a state machine, each transaction is an attempt to change the state, and each time the consensus generated block is the participant confirms the result of the state change caused by the transaction in the block [14,27].

In implementation, it is first assumed that there is a distributed data record book, which can be added and cannot be deleted. The basic structure of the bottom of the account is a linear linked list, which is also the source of its name "block chain". The linked list is composed of a series of blocks. The successor blocks record the hash (pre hash) of the leading block [28]. New data to be added must be put in a new block. And whether the block and transactions in this block are legitimate can be quickly checked by calculating the hash value. Any maintenance node can propose a new legal block. However, a consensus mechanism must be adopted to reach agreement on the final selected block.

3.2 Reed-Solomon Code

Reed-Solomon code is a linear coding method defined on the domain. The coding method generates k source data to generate l coded data, which is consistent with the FEC coding idea. In our system, the Reed-Solomon code is used to implement the FEC coding transformation of the packet layer [29].

When the user uploads electronic data, the system will first segment the electronic data. The fragmentation is mainly dependent on coding in Reed-Solomon code. Users need to provide two important parameters, the number of information fragments and the number of redundant fragments. Based on the above two parameters and the size of the files to be uploaded, the system adjusts the appropriate size of the encoding buffer to complete the fragmentation of the files [30].

Suppose the message length is d, there are n information data blocks, m redundant parity blocks.

(1) Divide the message D by word length = 8, and fill the missing part with 0 to get the data matrix. Get the data matrix $D = (D_1, D_2, ..., D_n)^{'}$.
(2) Generating an encoding matrix.

$$B = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ B_{11} & B_{12} & B_{13} & \cdots & B_{1n} \\ B_{21} & B_{22} & B_{23} & \cdots & B_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ B_{n1} & B_{n2} & B_{n3} & \cdots & B_{nn} \end{pmatrix}$$

(3) The matrix is multiplied by the coding matrix B and the data matrix D to obtain an encoded data matrix E.

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ B_{11} & B_{12} & B_{13} & \cdots & B_{1n} \\ B_{21} & B_{22} & B_{23} & \cdots & B_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ B_{m1} & B_{m2} & B_{m3} & \cdots & B_{mn} \end{pmatrix} \cdot \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{pmatrix} = \begin{pmatrix} D_1 \\ \vdots \\ D_n \\ C_1 \\ \vdots \\ C_m \end{pmatrix}$$

(4) Decoding process: delete the row corresponding to the missing piece of data from the coded slice and the coding matrix (assuming D_1 and C_2 are lost).

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ B_{11} & B_{12} & B_{13} & \cdots & B_{1n} \\ B_{31} & B_{32} & B_{33} & \cdots & B_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ B_{m1} & B_{m2} & B_{m3} & \cdots & B_{mn} \end{pmatrix} \cdot \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{pmatrix} = \begin{pmatrix} D_2 \\ \vdots \\ D_n \\ C_1 \\ C_3 \\ \vdots \\ C_m \end{pmatrix}$$

(5) Calculate the invertible matrix of B'.

$$(B')^{-1} \cdot B' \cdot D = (B')^{-1} \cdot E'$$

(6) Calculate the original message D, complete the encoding.

$$D = (B')^{-1} \cdot E'$$

3.3 Improved RFM Model

This paper uses the improved RFM model to score the storage nodes and use this score to achieve load balancing of distributed storage [31]. First, the storage node is specified to store an information slice for the first time to obtain 20 points. The gain obtained by storing one piece of information each time later is calculated by the following formula:

$$r_i = \begin{cases} \frac{20b_i}{B}, & b_i \neq 0 \\ 20, & b_i = 0 \end{cases}$$

where b_i indicates the number of storage slices of the storage node; B indicates the total number of storage slices of all storage nodes.

R indicator: The ratio of the last time the storage node stores the file to the total running time of the system.

$$R = \frac{t_1 - t_0}{T}$$

where t_1 indicates the time at which the storage node stores the current slice; t_2 indicates the time of the last slice storage; T indicates the total running time of the system.

F indicator: the average ratio of storage revenue per storage node to the system's specified revenue.

$$F = \frac{c}{aS}$$

where c indicates the total revenue of the storage node; a indicates the storage node storage fragment number; S indicates the revenue of the first specified storage of one slice (20 points)

M indicator: the storage node's storage revenue level of the points in all storage nodes of the system.

$$M = \frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{(c_i - \mu)^2}{2\sigma^2}]$$

where c_i indicates the total revenue of the storage node; n indicates the number of storage nodes; $\mu = \frac{1}{n}\sum_{i=1}^{n} c_i$ indicates the mean of the number of revenues of all storage nodes; $\sigma = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(c_i - \mu)^2}$ indicates the standard deviation of the revenue of all storage nodes;

This paper Uses FAHP to analyze the weights of R, F and M, the corresponding weights are as follows:

$$\vec{w} = [0.2296, 0.3459, 0.4245]$$

When performing storage node selection, calculate the RFM model score for each node using the following formula:

$$Score = \vec{w} \cdot \vec{m}$$

where $\vec{m}$ indicates the current R, F and M index value vector of the storage node.

In order to enable the storage node to implement load balancing of information slice storage, when the score is larger, the probability of being selected should be lowered, so the reciprocal is used to represent the final score.

$$Score_{final} = \frac{1}{Score}$$

3.4 Multi-target Node Decision Model

As an electronic data storage and proof system, the system relies on multiple user nodes to complete the storage and proof of electronic data. This process involves the selection of multi-user nodes, and we design how to select several of the best current nodes of the system and cooperate with the distributed storage work. Figure 2 shows the main design process of user node performance evaluation.

According to Ethereum, it relies on the computational power of each node to keep it running. In this system, it relies mainly on the storage power of each node. Therefore, it evaluates the performance of the user node from both network performance and storage level. More specifically, the network performance affects the transmission speed of the electronic data fragmentation, and the storage level affects the storage reliability of the electronic data fragmentation.

The parameter weight of the network performance is calculated by the fuzzy analytic hierarchy process (FAHP) [32] to calculate the weight of the bandwidth, network delay and packet loss rate, and the network performance is obtained. Then, the ranking of the user nodes is mainly calculated by the Three-base point method (TOPSIS) [33,34] to calculate the Euclidean distance between the ideal base point and the ideal superior base point and the inverse ideal base point, and the integrated distance is used to rank the nodes. (Before the storage node sorting operation, this paper has normalized the network performance, storage performance and RFM score.).

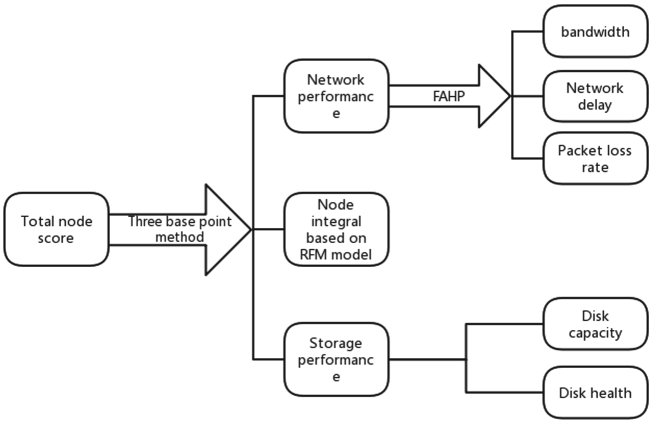

Fig. 2. Node evaluation framework

(1) Calculate the ideal superior base point and the inverse ideal optimal base point.
Assume that the maximum network performance is $s_1^{\triangle}$, the maximum storage performance is $s_2^{\triangle}$, and the maximum RFM score is $s_3^{\triangle}$ for all storage nodes.(n indicates the number of storage nodes; $k = 1, 2, 3$)

$$s_k^{\triangle} = \max\{s_{k_1}, s_{k_2}, \ldots, s_{k_n}\}$$

Assume that the minimum network performance is $s_1^{\triangledown}$, the minimum storage performance is $s_2^{\triangledown}$, and the minimum RFM score is $s_3^{\triangledown}$ for all storage nodes.

$$s_k^{\triangledown} = \min\{s_{k_1}, s_{k_2}, \ldots, s_{k_n}\}$$

(2) Calculate the distance between each point and the ideal superior base point and the inverse ideal base point.
In the m-dimensional space, it is not difficult to find that the excellent base points appear in the form of points. Using the Euclidean distance calculation method, we can calculate the distance ($d_i^{\triangle}$) between the excellent base point of the network performance, storage performance, the RFM score and the ideal superior base point, and the distance ($d_i^{\triangledown}$) from the inverse ideal optimal base point. ($i = 1, 2, ..., n$)

$$d_i^{\triangle} = \sqrt{\sum_{k=1}^{3} (z_{k_i} - s_k^{\triangle})^2}$$

$$d_i^{\triangledown} = \sqrt{\sum_{k=1}^{3} (z_{k_i} - s_k^{\triangledown})^2}$$

(3) Calculating the integrated distance.
By measuring the gap between the current plan and the optimal and worst case ideals, the pros and cons of the current programs are judged. According to the principle of minimum distance, the scheme corresponding to the point with the smallest integrated distance is selected as the optimal scheme.

$$\min\{d_i = \frac{d_i^{\triangle}}{d_i^{\triangle} + d_i^{\triangledown}}\}$$

4 System Design

We develop data sharing mechanism for data sharing based on block chaining to ensure data security and provenance. The detailed flow chart of the storage and proof system, as shown in Fig. 3, is divided into 6 main steps.

As shown in Fig. 3, the block chain storage system consists mainly of the proof user node, the storage user node and the smart contract, in which the node A can either be a proof user node or a storage user node. The system carries out a storage procedure as follows:

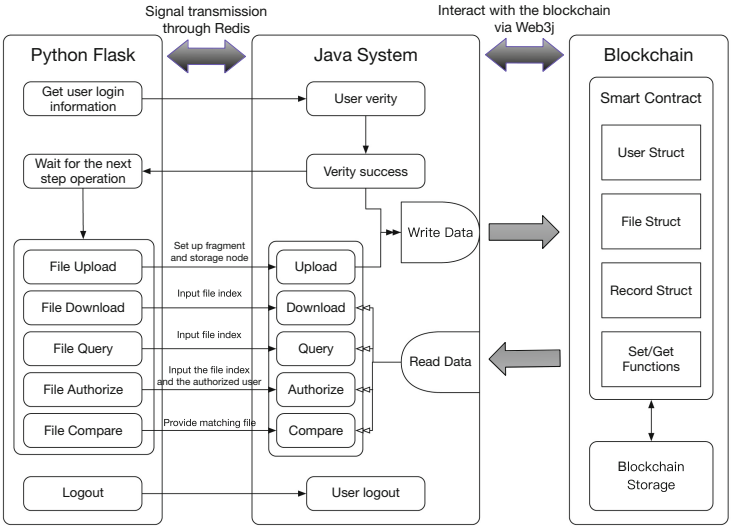

Fig. 3. Detailed flow chart of system

(1) After the user login system, the contract class member gets the user information and obtains the performance information of other nodes through the user node and obtains the value for the later data slice storage performance host computing.
(2) Files uploaded by users need to be stored in documents, and the key information of files is stored in the contract class file to form corresponding mapping relationship between data and users.
(3) The redundant fragment algorithm is used to slice the uploaded electronic data and select several optimal performance nodes according to the performance information of the nodes, which can be used to store each piece of data.
(4) According to the selected nodes, the system distributes data to different nodes, and returns the key information to the smart contract, including the IP address of the data storage, the absolute path of the data storage and the hash value of the data fragment in the record object.
(5) When the user needs to download or query the electronic data, the system puts forward the authorization request and establishes the corresponding mapping relation between the user information and the data needed to be accessed, then the user can carry on the related operation.
(6) After the user has access to the authority, it reads the relevant information of the electronic data from the contract. The system finds the location of the electronic data storage according to the storage information, downloads the electronic data and reduces the data and compares the hash value of the file to verify the integrity of the electronic data.

The main process of this system is to upload, save, view, download, compare, and authorize electronic data files. The process is introduced as follow.

4.1 System Upload Function

First, the user logs in to the system. When the user requests an electronic data upload operation, the system calculates the number of points m spent on the upload according to the number of electronic data fragments, and then calls the contract point function to obtain the current number of point c of the user. The system performs the judgment of the number of point.

After the judgment, it provides the upload function to the user, and updates the point information corresponding to the user on the blockchain. If the judgment is not passed, the user is informed that he/she does not have enough points.

4.2 System Data Preservation Function

First, the system uses Reed-Solomon code to divide the electronic data into redundant data and obtains n data sheets and m redundant slices. The system uses iperf to obtain the performance parameters of the system storage host, mainly including throughput, delay, and bandwidth, and calculates the network performance by calculating three parameters based on fuzzy analytic hierarchy process.

At the same time, the system uses smart to get the hard disk capacity and hard disk health of the storage host. The network performance score and disk storage capacity are used as parameters, and the ideal base point method is used to calculate the comprehensive distance of the performance of the storage host and sort it to obtain the optimal performance hosts.

The system randomly distributes the fragments of the electronic data to the selected hosts, fixes them into the storage area of the blockchain through the consensus algorithm of the blockchain, and writes the fragmentation information into the blockchain.

4.3 System Data Query Function

The user can use the electronic data query function at any time to obtain the stored information with the file. According to the need for electronic data storage and proof, the data is only visible to uploaders by default. When users need to view the electronic data of other users, they need to obtain the authorization of the corresponding user. When the user makes a query request, the system interacts with the contract, determines the user authority and judges whether the file belongs to the user node.

After the success, the index number corresponding to the data file and the index number of the fragment storage location are read. Then, it is determined whether the file index is smaller than the number of user files, and the absolute index bit of the file is obtained. The information (IP address and stored absolute path) is stored in the blockchain and the electronic data is found through the information.

4.4 System Data Download Function

When the user submits the download request to the system, the automatic downloading and decoding of the file is realized after locating the corresponding data information in the block chain storage area and finding the location of the data storage. This is similar to data query function. Data files that users can download must be uploaded by themselves or authorized by others to ensure users' privacy and security.

4.5 System Data Comparison Function

After getting the data returned, the system calls the SHA-1, SHA-256 and MD5 hash algorithms respectively, and compares the hash values of each slice to the hash value stored in the smart contract. If the hash value is equal, it means that the file has not been tampered with. When the hash value of part of the system is not equal to the storage in the contract, if the number of unequal fragments is less than the number of redundant pieces of the system data, the system can still restore the source file.

4.6 System Data Authorization Function

Due to the needs of the system, the user's personal privacy needs to be protected, so the default user's electronic data is their personal information file. Users need to authorize others, so that other users can have access to the authorized documents and electronic data. In authorization, the user's public key is input, that is, user account and the serial number of the authorized file. The system will write the authorized file information to the blockchain, then the authorization ends.

4.7 The Distributed Storage Architecture Diagram of File Fragmentation

The biggest advantage of distributed system is that it enhances the fault tolerance and balances the different elements of the same type in the entire system [26,35]. We run the system program on several servers and use Nginx server to achieve load balancing. When users access files through domain names, the Nginx server receives HTTP requests from the interconnected files. It will select the appropriate system server according to the server connection situation and forward the HTTP requests from the users to the selected system server.

Since each system server can communicate with each other, when file distribution is performed, the fragments are distributed to each system server according to the distribution policy. And each system server carries a geth client, which connects to the same private chain through IPC files. The multi-node private chain architecture ensures that file information generated by the system is stored in the same chain, as shown in Fig. 4.

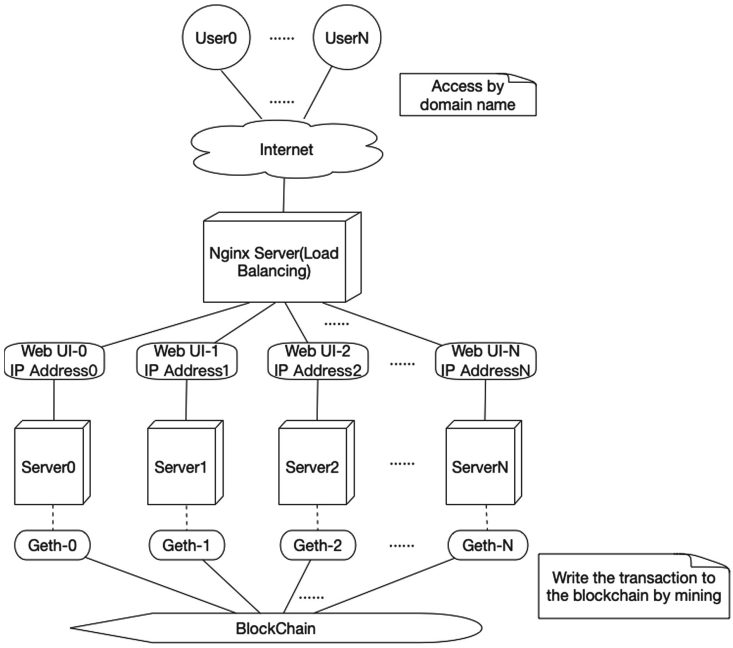

Fig. 4. The distributed storage architecture diagram

This section introduces the implementation of the electronic storage and proof system based on blockchain intelligent contract, and describes the data uploading, preserving, querying, downloading, verifying, authorizing and the interaction with the system. Among them, the intelligent contract part is the underlying data storage method to support the system. File uploading, fragmentation, storage, downloading, and verification are the main functional points of the system.

5 System Function Test and Evaluation

Before providing a systematic test evaluation, we summarize these capabilities of the system:

(1) The system provides real-time auditing of all data accesses in the storage application. We use electronic data files as data units, audit all operations on the data objects, and record using blockchains. In this way, all electronic data access situations can be collected and monitored.
(2) For each piece of electronic data, we upload the data to the blockchain network. By doing so, we create an unchangeable file data fingerprint, and the system has secure and permanent record keeping and tamper-proof timestamps. Any changes to system data are detected by verifying blockchain data comparisons.

(3) Users can view data services while protecting their privacy. User access records are anonymous in the blockchain network. The data source is unable to query the user account. Anonymous saving is reflected in two aspects: on the one hand, because the user ID is hashed randomly, the user identity is not connected to the source data. On the other hand, non-connectivity between each user is also achieved, especially for uploading user protection of authorized data.

5.1 The Analysis of System File Upload Performance

Keep the number of fragments unchanged, change the file size.

Before performing performance tests, we make some files of specified size. To analyze the impact of file size on upload performance more easily, it is necessary to keep the number of fragments unchanged. The number of fragments in this experiment is set to 5. According to the file size increasing order, the file encoding and fragmentation in the uploading process, the storage server selection time, the fragment distribution time, and the total upload time are counted, which is shown in Table 1.

Table 1. File upload duration (5 slices)

	1.2 Mb	2.4 Mb	12.4 Mb	59.0 Mb	108 Mb	190 Mb
File encoding and fragmenting duration(s)	3.584	3.706	4.189	6.852	10.302	19.538
Server selection duration(s)	0.490	0.439	0.424	0.417	0.416	0.438
Fragment distribution duration(s)	10.631	10.516	10.596	10.935	11.427	11.904
Total duration(s)	14.705	14.661	15.209	18.204	22.145	31.880

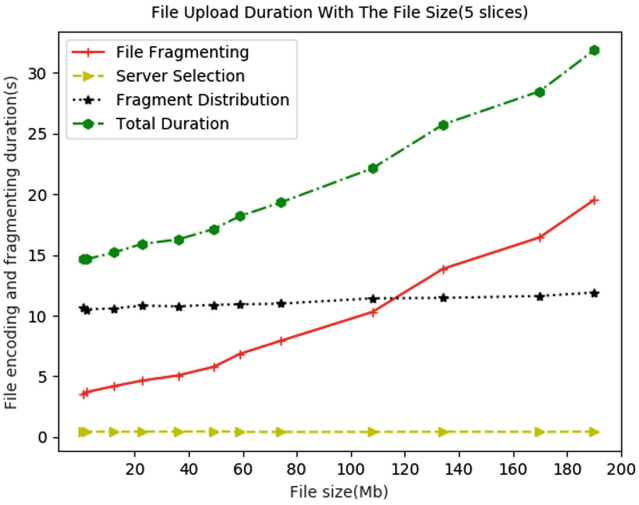

Fig. 5. File upload duration with the file sizes

Based on the data obtained, the line graph more directly shows the trend of the cost of the file during the various stages of the upload process, as shown in Fig. 5. It can be easily seen from the figure that the file size has a great influence on the time of file encoding and fragmentation, whereas it has little effect on the time and fragment distribution of the storage server selection. Therefore, it can be concluded that the file encoding fragmentation time increases with the file size, which lead the total upload time to increase.

Keep the file size unchanged, change the number of fragments.

By fixing the file size to 19.8 Mb and changing the number of fragments, we obtain the statistics in Table 2.

Table 2. File upload duration (File size is 19.8 Mb)

	3	6	9	15	24	30
File encoding and fragmenting duration(s)	4.153	4.174	4.447	4.264	4.668	4.724
Server selection duration(s)	0.455	0.453	0.405	0.437	0.408	0.405
Fragment distribution duration(s)	6.685	12.687	18.775	30.736	49.619	61.315
Total duration(s)	11.293	17.314	23.627	35.437	54.695	66.444

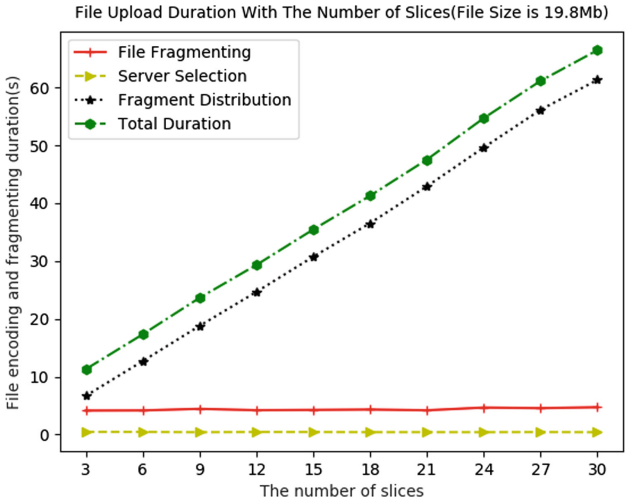

Fig. 6. File upload duration with the number of slices

Similarly, we show the trend of change as a line chart in Fig. 6. It can be easily seen from the figure that the number of fragments has a great influence on the time of fragment distribution, which exhibits a linear variation characteristic. However, it has little effect on the duration of storage server selection and file encoding. Therefore, it can be concluded that the time of fragment distribution increases with the number of fragments, so that the total upload time increases.

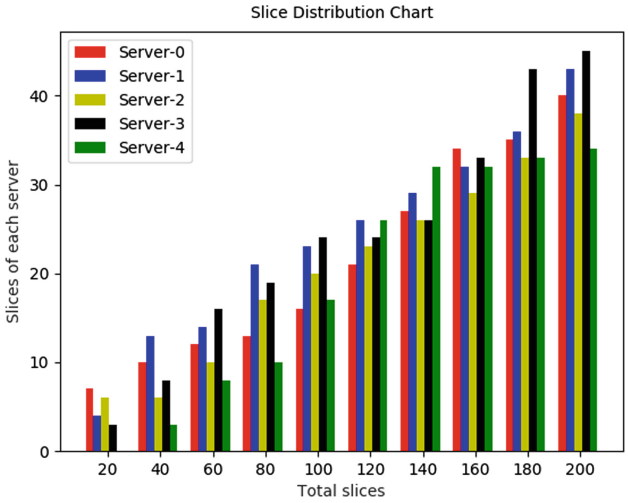

Fig. 7. Slice distribution

5.2 The Balance Test of File Distribution Strategy

For the distribution of file fragmentation, we combine the integration model and server performance to comprehensively evaluate the storage server situation, and select the corresponding server to receive file fragmentation by comprehensive evaluation score. In theory, as the total number of slices in the system increases, the number of slices on each storage server should be similar. It would be unexpected if some server utilization is too low. To verify our expectation, we prepare 5 cloud servers for deployment, constantly change the number of total slices in the system. We count the number of slices owned by each storage server and plot it as a composite bar chart shown in Fig. 7.

As is directly depicted by the graph, the fragment distribution meets our expectations comparatively, which further verifies that the fragment distribution strategy is reasonable.

6 Conclusion

We mainly record the electronic data storage in a decentralized storage and proof system on the blockchain based on the non-tampering feature of the it. First, we introduce the research background and significance of electronic data storage and proof. Then we draw relevant solutions according to its existing problems. Subsequently, we investigate the status of research and development in the world and use it to develop our system. Secondly, we introduce the main functional architecture of the system, the theoretical knowledge involved in the system, and explain the related technologies in the architecture. Then the system total process is analyzed, and the detailed requirements analysis of the main

functional modules of the system is carried out, while the key points involved in the system improvement function are clarified. On this basis, we use the latest blockchain technology and distributed storage technology to design and implement a blockchain-based electronic data storage and proof system, which is illustrated by the system graphics module.

With the emerging technology of blockchain, we anchor the key "digital fingerprint" of electronic data in the storage area of blockchain, and combine technologies such as intelligent contract, distributed storage, fault-tolerant coding, and multi-attribute decision making. We realize the electronic data storage and proof system based on blockchain. The system ensures the authenticity, integrity, and uniqueness of electronic data by making use of the core features of the block chain's decentralization and non-tampering. At the same time, the system fully considers the fault-tolerant requirements of the distributed storage system and uses the Reed-Solomon code to protect the electronic data redundantly, which reduces the problems caused by the failure of a single server or the transmission channel. In addition, the system has also developed a point system for users to ensure that the system can attract more users to join, and thus improve the reliability of this storage and proof system.

Currently, we have developed a simple blockchain-based electronic data storage and proof system. The interaction between the hosts of the system is only carried out in the local area network. In the future, the system is further optimized to realize the electronic data storage in the WAN. The formula algorithm used in the blockchain of this paper is proof of workload. The execution time of this algorithm is long, and there is a waste of system resources. The current consensus mechanism algorithm can be optimized later. In the process of communication between the client and the server in the system, the plain text communication method is adopted, which has potential security risks, and symmetric encryption can be used later to improve security.

References

1. Liao, D.Y., Wang, X.: Design of a blockchain-based lottery system for smart cities applications (2017)
2. Shae, Z., Tsai, J.J.P.: On the design of a blockchain platform for clinical trial and precision medicine. In: IEEE International Conference on Distributed Computing Systems (2017)
3. Xu, R., Lu, Z., Zhao, H., Yun, P.: Design of network media's digital rights management scheme based on blockchain technology. In: IEEE International Symposium on Autonomous Decentralized System (2017)
4. Yue, X.: Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. J. Med. Syst. **40**(10), 218 (2016)
5. Ouaddah, A., Elkalam, A.A., Ouahman, A.A.: Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In: Rocha, Á., Serrhini, M., Felgueiras, C. (eds.) Europe and MENA Cooperation Advances in Information and Communication Technologies. Advances in Intelligent Systems and Computing, vol. 520, pp. 523–533. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-46568-5_53

6. Zyskind, G., Nathan, O., Pentland, A.S.: Decentralizing privacy: using blockchain to protect personal data. In: IEEE Security & Privacy Workshops (2015)
7. Cheng, J.C., Lee, N.Y., Chi, C., Chen, Y.H.: Blockchain and smart contract for digital certificate. In: 2018 IEEE International Conference on Applied System Invention (ICASI) (2018)
8. Pilkington, M.: Blockchain Technology: Principles and Applications. Social Science Electronic Publishing, Rochester (2015)
9. Qi, X., Sifah, E.B., Asamoah, K.O., Gao, J., Guizani, M.: MeDShare: trust-less medical data sharing among cloud service providers via blockchain. IEEE Access **5**(99), 14757–14767 (2017)
10. Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., Njilla, L.: ProvChain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID) (2017)
11. Schwerha, J.J.: Cybercrime: legal standards governing the collection of digital evidence. Inf. Syst. Front. **6**(2), 133–151 (2004)
12. Hardjono, T., Smith, N., Pentland, A.S.: Anonymous Identities for Permissioned Blockchains (2016)
13. Wu, F., Pai, H.T., Zhu, X., Hsueh, P.Y., Hu, Y.H.: An adaptable and scalable group access control scheme for managing wireless sensor networks. Telematics Inf. **30**(2), 144–157 (2013)
14. Dinh, T.T.A., et al.: BLOCKBENCH: A Framework for Analyzing Private Blockchains (2017)
15. Cachin, C., Vukolić, M.: Blockchain Consensus Protocols in the Wild (2017)
16. Kakavand, H., Nicolette, K.D.S., Chilton, B.: The Blockchain Revolution: An Analysis of Regulation and Technology Related to Distributed Ledger Technologies. Social Science Electronic Publishing (2016)
17. Kuo, T.T., Kim, H.E., Ohno-Machado, L.: Blockchain distributed ledger technologies for biomedical and health care applications. J. Am. Med. Inf. Assoc. **24**(6), 1211–1220 (2017)
18. Stanciu, A.: Blockchain based distributed control system for edge computing. In: International Conference on Control Systems & Computer Science (2017)
19. Dorri, A., Steger, M., Kanhere, S.S., Jurdak, R.: BlockChain: a distributed solution to automotive security and privacy. IEEE Commun. Mag. **55**(12), 119–125 (2017)
20. Herlihy, M.: Blockchains and the future of distributed computing. In: ACM Symposium on Principles of Distributed Computing (2017)
21. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the Internet of Things. IEEE Access **4**, 2292–2303 (2016)
22. Cong, L.W., He, Z.: Blockchain Disruption and Smart Contracts. Social Science Electronic Publishing, Rochester (2018)
23. Xu, X., et al.: A taxonomy of blockchain-based systems for architecture design. In: IEEE International Conference on Software Architecture (2017)
24. Peters, G.W., Panayi, E.: Understanding modern banking ledgers through blockchain technologies: future of transaction processing and smart contracts on the internet of money. In: Tasca, P., Aste, T., Pelizzon, L., Perony, N. (eds.) Banking Beyond Banks and Money. NEW, pp. 239–278. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42448-4_13
25. Sharples, M., Domingue, J.: The blockchain and kudos: a distributed system for educational record, reputation and reward. In: Verbert, K., Sharples, M., Klobučar, T. (eds.) EC-TEL 2016. LNCS, vol. 9891, pp. 490–496. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45153-4_48

26. Shi, E., Shi, E.: FruitChains: a fair blockchain. In: ACM Symposium on Principles of Distributed Computing (2017)
27. Suzuki, S., Murai, J.: Blockchain as an audit-able communication channel. In: Computer Software & Applications Conference (2017)
28. Khalil, R., Gervais, A.: Revive: rebalancing off-blockchain payment networks. In: ACM SIGSAC Conference on Computer & Communications Security (2017)
29. Huo, Y., El-Hajjar, M., Maunder, R.G., Hanzo, L.: Layered wireless video relying on minimum-distortion inter-layer FEC coding. IEEE Trans. Multimed. **16**(3), 697–710 (2014)
30. Glaser, F.: Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain Enabled System and Use Case Analysis. Social Science Electronic Publishing, Rochester (2017)
31. Dudhia, A.: The reference forward model (RFM). J. Quant. Spectrosc. Radiat. Transf. **186**, 243–253 (2017)
32. Sadovykh, A., Hein, C., Morin, B., Mohagheghi, P., Berre, A.J.: An MAGDM based on constrained FAHP and FTOPSIS and its application to supplier selection. Math. Comput. Model. **54**(11), 2802–2815 (2011)
33. Shih, H.S., Shyur, H.J., Lee, E.S.: An extension of TOPSIS for group decision making. Math. Comput. Model. **45**(7), 801–813 (2007)
34. Yu, Z., Wen, J.: The IoT electric business model: using blockchain technology for the Internet of Things. Peer Peer Networking Appl. **10**(4), 983–994 (2017)
35. Aniello, L., Baldoni, R., Gaetani, E., Lombardi, F., Margheri, A., Sassone, V.: A prototype evaluation of a tamper-resistant high performance blockchain-based transaction log for a distributed database (2017)