Research Article

A Big Data Security Architecture Based on Blockchain and Trusted Data Cloud Center

Peng Qin,¹ Wei Li,¹ and Ke Ding²

¹School of Computer Science, Liupanshui Normal University, Liupanshui 553004, China
²School of Economics and Management, Liupanshui Normal University, Liupanshui 553004, China

Correspondence should be addressed to Peng Qin; qinpengcumt@126.com

Received 20 April 2022; Revised 25 May 2022; Accepted 20 July 2022; Published 31 August 2022

1. Introduction

The development of the Internet has provided a convenient lifestyle for people, and more and more social activities have been networked. There have a large number of sensitive data generated in social activities, such as personal information, financial data, electronic health records, travel data, and network activities, which are significant for individuals. So, some scholars focus their research on the field of how to safely store and effectively access data stored in the cloud.

Cloud computing technology provides service support for the storage and analysis of big data. More and more data and applications are stored in the cloud, enabling real-time access and sharing. At the same time, many data security and cloud data leakage issues have arisen. For example, in 2015, the Wanda Group’s official website APP cloud suffered an SQL injection attack, which resulted in the leakage of users’ sensitive information. In July 2018, all the platform data stored in the cloud by cutting-edge CNC companies were lost. According to the report of the third quarter of 2019 by the Risk Based Security (RBS), from January 1, 2019 to September 30, 2019, there were 5,183 data leakage incidents disclosed worldwide, and the leaked data volume reached 7.995 billion records.

The main contribution of this paper is to comprehensively analyze the existing big data security storage schemes, propose a new big data security architecture based on blockchain and trusted data cloud center, introduce the implementation process of the scheme in detail, and conduct experimental analysis. The scheme improves the data storage and access strategy, strengthens the data access control, improves the data access efficiency, and ensures the data security through the comprehensive application of trusted authentication, intrusion detection, data segmentation, hierarchical storage, and other technologies. The innovation of the scheme is the introduction of PairHand user authentication association, blockchain technology, and access mode based on user authority, which realizes decentralized storage, user trusted authentication, and convenient access.

The paper is structured as follows. Section 2 introduces related research on big data security architecture in cloud storage. In Section 3, a big data security architecture based on blockchain and trusted data cloud center was proposed. Section 4 is scheme analysis, which includes safety analysis.
and data availability analysis. Finally, conclusions and future work are presented in Section 5.

2. Related Work

Many scholars had made active research on big data security protection in the cloud computing environment. Thayananthan and Albeshi [1] proposed a big data security protection scheme based on a mobile data center, which adopted the idea of hierarchical design, in view of quantum cryptography and PairHand user authentication protocol, to realize the security of cloud platform, cloud data, and cloud applications. However, the scheme lacked the hierarchical storage, data backup, and risk sharing of data.

Manogaran et al. [2] put forward a MetaCloudDataStorage scheme, which is a security structure in cloud computing. Data is divided in the meta cloud center and stored in different levels of cloud. The cloud is responsible for ensuring the security of data. The metacloud center stores data segmentation and physical information of actual storage. The method lacks authentication and security detection of data access, and the cloud data is not encrypted too.

Sookhak [3] proposed a big data cloud protection solution based on remote data audit RDA technology, which uses remote data signature, data redundancy reduction, and data split table to realize rapid integrity verification and dynamic data operation of data, but the scheme lacks authentication between the cloud and data provider.

Hanshu and Zhixin [4] proposed a cloud storage security model based on data flow, which realized data security access and storage, as well as data integrity verification. However, in this model, the third-party authentication center lacks trusted authentication, and data encryption increases data processing workload, which is easy to cause communication congestion.

Jingjiang [5] proposed a security risk assessment model based on the cloud life cycle, which predicted the risks from various steps and services in the cloud environment, and simulated in a specific experimental environment, but the assessment model is not universal and can only be limited to specific areas.

Kumar et al. [6] put forward an identity based big data security authentication framework, which adopts the elliptic cryptography algorithm to reduce user identification and realizes mutual authentication of identity through servers distributed near users. The advantages of the model are convenient authentication, weak antiattack, and easy forgery of server identity.

Ruijin et al. [7] proposed a block chain data sharing model based on ring signature based on the idea of decentralized storage to protect the privacy of data users but did not provide corresponding consensus mechanism and reward mechanism.

Li [8] proposed a blockchain-based big data model bcBIM. It guarantees to trace, authenticate, and prevent tampering historical data. Certainly, it also can generate a unified format to support data sharing, data audit, and data provenance. However, this model is mainly used to analyze the building information modeling.

Alhazmi et al. [9] proposed a big data security framework by using blockchain technology and combining fragmentation, encryption, and access control technology to encrypt and protect all process data and realize data security protection. The disadvantage of the scheme is that it has high computational overhead and does not provide an interface for users to access. The relevant literature is summarized in Table 1.

In view of the shortcomings in the above schemes, this paper designs and implements a data security architecture based on blockchain and trusted data cloud center, which adopts the ideas of trusted authentication, intrusion detection, data segmentation, and decentralized storage, and applies Amazon Web Services log processing and key management services, PairHand user authentication protocol, Hadoop data analysis framework, and hive data query. It realizes data security protection and fast legal access in the cloud environment.

3. A Big Data Security Architecture Based on Blockchain and Trusted Data Cloud Center

3.1. Scheme Framework. The entities in the big data security framework based on block chain and trusted data cloud center include data owner, service provider, cloud application interface, data cloud center DCC, data storage cloud service provider, and security service [10]. The whole data security architecture adopts the idea of hierarchical design. The cloud interface is at the top level responsible for processing user requests. The data cloud center is in the core layer, which is responsible for the business logic of the entire data processing, including data segmentation, hierarchical storage, data path saving, and key certificate management. The data storage cloud blockchain is located at the bottom, which is responsible for specific data storage and backup. Security services are mainly focused on data cloud center and storage services, responsible for certificate authentication, intrusion detection, and data encryption [11, 12]. The framework as shown in Figure 1.

The whole data flow process is based on the authentication between the service provider and DCC, and data interaction can only be carried out after the authentication of each other. In this process, data users need to register in advance, and the users who complete the registration have access to the data.

3.2. Method of Implementation

3.2.1. PairHand System Initialization and User Authentication [2, 11]

(i) Initialization: let $G$ be a cyclic addition group of order $q$ and $G_1$ be a cyclic multiplication group of order $q$, $e^* : G \leftrightarrow G_1$, the data cloud center DDC selects a random number $s$ as the key and calculates the public key $P_{pub} = sp$, select two secure hash functions $H_1$ and $H_2$, where $H_1 : \{0, 1\} \rightarrow G$ and $H_2 : \{0, 1\} \rightarrow \mathbb{Z}_q^*$, the data cloud center announces $\{G, G_1, q, P, P_{pub}, H_1, H_2\}$, and save the
private key $s$. For each user, DDC calculates the corresponding public key and private key, the public key is $H_1(\text{ID}_{\text{user}})$, and the private key is $sH_1(\text{ID}_{\text{user}})$.

(ii) User registration: the data owner or service provider is registered in data cloud center DCC through ID\_user. DCC verifies user identity and selects user random identity series PID\{pid\_1, pid\_2...pid\_j\}, calculates the public key and private key pair for the identity, and sends the (pid\_i, sH\_1(pid\_i)) tuple to the user securely. A valid time is added to pid\_i, after which the corresponding identity and private key become invalid.

(iii) User authentication:

(1) When users request service from DCC, they randomly select an unused identity pid\_i and the corresponding private key sH\_1(pid\_i), calculate the signature $\sigma_i = H_2(M_i) \cdot sH_1(\text{pid}\_i)$, and send the message and signature to DCC, where $t_i$ is the timestamp, and ID\_DCC is the data cloud center identity.

(2) Users calculate symmetric key $K_i$ shared with DCC, $K_i = e^{sH_1(\text{pid}\_i), H_1(\text{ID}\_\text{DCC})}$

(3) After receiving the message and signature, DCC performs the following steps:

First step: checking timestamps $t_i$ and service expiration time of the pid\_i.
Second step: verifying signature validity: \( e^h(\sigma_i, P) = e^h(H_2(M_i) \cdot sH_1(pidi), P_{pub}) \)

Third step: calculating the symmetric key \( K_i, K_i = e^h(H_i(pidi), sH_1(pidi), H_i(ID_{DCC})^s = e^h(H_1(pidi), sH_1(pidi), H_1(ID_{DCC})) \)

Fourth step: calculating authentication code \( Aut = H_2(K_i||pidi||ID_{DCC}) \), then sending Aut and temporary session token \( TK_{ts} \) to the user. After receiving the Aut, the user uses \( K_i \) to calculate the verification code \( Ver = H_2(K_i||pidi||ID_{DCC}). \) If equal, mutual authentication between the user and the data cloud center is realized. Data transmission and service requests can be performed by the temporary session token \( TK_{ts} \)

(iv) After receiving the user request with \( TK_{ts} \), DCC initiates the data storage or service request according to the demand. The authentication process of users and data cloud centers as shown in Figure 2

3.2.2. Cloud Application Interface. The cloud application interface is mainly responsible for receiving user requests, verifying user identity, providing channels for data owner and legitimate users to access data and services, and passing data requests to cloud data centers, shielding specific processing logic, and receiving the returned results of the cloud data center.

3.2.3. Data Processing of DCC

(1) Data Storage. After DCC receives the user’s data storage request, according to the user’s definition, the data is stored hierarchically according to three levels: sensitive, important, and common. Different levels of data are stored by different encryption methods. Ordinary data only encrypts the data digest, and the contents are transmitted in clear text. Important data are randomly encrypted by symmetric encryption algorithms such as AES and DES, and sensitive data are completely encrypted by quantum encryption [13].

DDC realizes decentralized storage of data through blockchain technology. As a miner node in the block, each storage service provider obtains data storage rights and rewards through random number election mechanism [9]. A complete data is randomly divided into \( n \) parts, and each part can be represented as part, \( (i, (1, n)) \) and stored in \( m \) different cloud miner nodes. Each data storage generates a new block. The block header records the hash value, timestamp, random number of the previous block. The block body records the hash value of the storage resource, resource storage address, resource backup address, node information and searchable encrypted index. In order to ensure the security of data, the identity information of the cloud server is recorded in each block. The cloud server needs to make backup storage after obtaining access rights, and the backup path is reserved in the block body [14]. The blockchain storage structure of data as shown in Figure 3.

In order to ensure antiaffection of blockchain, the number of cloud service providers is more than 10, and they are distributed in different regions. DDC data cloud center is responsible for data partitioning and recording block mapping MapStoragePath of data storage.

\[
\text{MapStoragePath} = \{\text{Data}((P1(M1, M2 \ldots .Mr)) P2(M1, M2 \ldots .Ms)) \ldots Pn(M1, M2 \ldots .Mt))\}.
\]

In formula (1), \( P \) is the cloud miner node of actual storage, \( M \) represents the storage medium, and DDC encrypts the MapStoragePath. At the same time, DDC performs a hash operation on the entire file data and saves the corresponding hash value. When the data is obtained from the data storage cloud next time, the hash value should be calculated again. If it is the same, it means that the data is complete; if it is different, it means that the data is attacked [15].

All data are stored in different cloud service provider nodes, and each miner node has different fragment keys to ensure the reliability and security of data. Cloud service providers in the scheme need to provide corresponding data backup and save the data recovery path to ensure high data disaster tolerance [9, 10, 16].

(2) Service Request. When DCC receives the user’s service request, it will analyze the log information through AWS Cloud Trail to achieve intrusion detection in the beginning [17]. AWS analyzes information such as user identity, user permissions, last login time, current status, login location, and login IP to detect whether they are illegal users. If it is an illegal user, add the relevant information to the threat update table Threat_update; otherwise, accept the user’s request and provide services for the user.

3.2.4. User Security Access. In traditional data storage structures, data owners accessed data and services through authentication. In cloud computing, most data and services are stored in the cloud. To access data and services, ordinary users need to securely access, that is, pass cloud service authentication. In traditional secure access, users need to make an access request to the data owner or service provider and access the cloud service through the obtained certificate.
and key. This method requires the data provider to be online in real time. This article uses an access method based on user permissions [18]. The data owners and service providers have a permission operation table that stores users and users’ permissions for a certain file. When uploading data to the cloud, the permission operation table is also uploaded at the same time. When a user accesses the cloud, the cloud judges the user’s request according to the user permission table and provides the corresponding service response. If the request exceeds the user’s permission, the cloud will reject the service. If the request is within the permission range, the cloud will respond to the request. The user access model as shown in Figure 4.

3.3. Data Analysis

3.3.1. Log Analysis. In the scheme, MapReduce big data analysis and processing framework is adopted. To explore DCC daily user visits, user data requests, and abnormal data processing by analyzing log information, MapReduce is an open source big data processing framework, which mainly includes Map stage and Reduce stage. In the Map stage, the main node decomposes the tasks and divides the input data into different task nodes. The task nodes are responsible for counting the generated key-value pairs < key, value >. In the map stage, the same key-value pairs are counted and output [19]. Taking statistics of users’ daily data requests as an example, Figure 5 is an illustration of the process of processing log files by MapReduce framework.

3.3.2. Apply Apache Hive in AWS to Process Big Data. Apache Hive is a data warehouse platform based on Hadoop that facilitates ETL work and implements data extraction, conversion, and loading from source to destination. Hive provides a query language similar to SQL and HQL, which can convert the query language written by users into corresponding MapReduce programs for execution are based on Hadoop [20]. In this article, Apache Hive is mainly used to process log files stored in Amazon S3, store data to AWS, and extract data from AWS. Hive commands translate into a set of mapping Map and Reduce jobs that are executed on a cluster of computers for distributed data sets.

4. Scheme Analysis

4.1. Security Analysis

4.1.1. Trusted Authentication. Compared with the traditional secure cloud protection scheme, the design scheme in this article starts with user authentication and realizes mutual authentication between the user and the data cloud center. The PairHand solution is applied to the authentication process, adding time stamps and random identities, which can fully guarantee user identity privacy and forward security of data transmission and avoid significant losses caused by identity leakage.

4.1.2. Intrusion Detection. When a user requests service from DDC, the AWS-based log processing program performs user analysis based on IP, location, access time, frequency, and service request type. For illegal users, log processing program will reject user service requests and add them to the intrusion access table to achieve intrusion detection.

4.1.3. Decentralized Storage. The scheme adopts the idea of decentralized storage, in which the data is partitioned and stored in different cloud block nodes. The data cloud center only stores the block mapping information of data storage, which on the one hand reduces the data processing burden of the cloud center, and on the other hand increases the difficulty for illegal intruders to obtain data. Even if an intruder completely breaks through a certain cloud block, the data is
random and belongs to independent carriers of different sources. As long as the size of the data block is well controlled, the data can be fully secured. Different levels of data are saved in different ways. The quantum encryption algorithm is adopted for high-level sensitive data, which can fully ensure data security.

In the blockchain network, random number mechanism is used to select miners’ nodes and obtain storage rights and rewards. Time stamps are stored in the block, which can guarantee the irreversibility and traceability of transactions. At the same time, the block holds the hash value of resources, which can guarantee the integrity of each data and the privacy of users.

Assuming that the date is divided into $N$ copies by DCC and stored in $M$ different cloud service providers, each service provider stores $N/m$ copies of data. At the same time, assuming that the total number of cloud service providers is $M$, the probability of selecting $m$ service providers is $m/M$. Data stored in different service providers are randomly; so, the probability of data being completely attacked is $P$:

$$P = \frac{m}{M} \cdot \frac{1}{(N/m)!} \cdot \frac{1}{m!} = \frac{m}{MN!}.$$  \hspace{1cm} (2)

When $M$ and $N$ are large enough, data security can be guaranteed.

4.2. Detection Rate of Corrupted Data. In order to analyze the detection rate of the damaged data, the data blocks are set to perform experimental statistics on the detection rate from 1 to 3000, and the results as shown in Figure 6. When there are fewer corrupted data blocks, the detection rate is lower. When there are more corrupted data blocks, the detection rate is higher, and all detections are basically realized.

4.3. Data Availability Analysis. In the security architecture, users adopt the access method based on permission. As long as the user is registered with the data user to obtain file permissions, the application for access can be made through the data cloud center application interface. So, it avoids the situation that the data user cannot access the data when offline, which changes the traditional user access mode and improves the availability of data.

4.4. Efficiency Analysis. In this scheme, the data storage mapping stored in the trusted data cloud center is completely encrypted, and the actual physical data is stored in cloud blockchain. Different levels of data storage have different physical forms. Compared with the traditional data encryption method, it saves a lot of computational overhead and has high data processing efficiency [21].

The solution adopts the big data analysis architecture Hadoop and the idea of cloud computing, which fully guarantees the response time of the service. Data integrity verification is implemented in the data cloud center, which reduces data flow and frequent data interaction, ensures data
integrity, and improves data response speed. Compared with traditional authentication schemes, the legal access time of users is greatly reduced.

In order to study the influence of different data partition sizes \( k \) on data computational complexity, different \( k \) values were set to test calculation time, and the result as shown in Figure 7. When the amount of data is small, the size of data partition \( k \) has little influence on the computational complexity, but with the increase of data block, the larger the value of \( k \), the smaller the computational load.

In this paper, the blockchain adopts the random election consensus algorithm based on the calculation of the random number stored in the predecessor block, the hash value of the current block, and other nodes’ public keys to select miners’ nodes [22, 23], which reduces the information transmission from point to point. Compared with the traditional proof of Proof of Work workload and Proof of Stake, consensus efficiency is improved, and new nodes can be generated for data storage quickly. The comparison of time efficiency of the three consensus mechanisms is shown in Figure 8.

5. Conclusions and Future Work

The storage and access of big data need to balance time efficiency and data security and handle data backup and data recovery. In this paper, based on blockchain and trusted data cloud center, a big data security cloud architecture scheme with trusted authentication, data partitioning, and decentralized storage is proposed. The innovation of the scheme is to use PairHand to achieve user authentication, adopt the idea of blockchain decentralization to achieve data storage, apply quantum encryption to cloud storage and transmission, and achieve data analysis through AWS to fully guarantee the security and reliability of data, which has a good reference value for the security protection of big data cloud in the future.

The deficiency of this paper lies in the insufficient analysis and application of data, and the mainstream data mining technology is not used to mine the hidden value of big data. At the same time, due to the lack of intelligent analysis, there is insufficient research on dimensionality reduction processing and feedforward training of big data. In the future research, the industry data will be analyzed and processed. For different industry applications, different feature selection and feature extraction technologies will be adopted to realize the dimensionality reduction and storage of data. At the same time, artificial intelligence, data mining, and data visual analysis technologies will be used to reveal the patterns between data attributes, realize the mining and presentation of data value, and provide necessary reference for enterprise decision-making.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no competing interest.

Acknowledgments

The work was supported by the Youth Science and Technology Talent Growth Program of the Education Department of Guizhou Province (Grant No. KY 2018 [385]), Liupanshui Science and Technology Project (Grant No. 52020-2018-04-04), and Key Natural Science Projects of Liupanshui Normal University (Grant No. LPSSYZDK202204).

References


