Differentially Authorized Deduplication System Based on Blockchain

Abstract: In architecture of cloud storage, the deduplication technology encrypted with the convergent key is one of the important data compression technologies, which effectively improves the utilization of space and bandwidth. To further refine the usage scenarios for various user permissions and enhance user's data security, we propose a blockchain-based differential authorized deduplication system. The proposed system optimizes the traditional Proof of Vote (PoV) consensus algorithm and simplifies the existing differential authorization process to realize credible management and dynamic update of authority. Based on the decentralized property of blockchain, we overcome the centralized single point fault problem of traditional differentially authorized deduplication system. Besides, the operations of legitimate users are recorded in blocks to ensure the traceability of behaviors.

Keywords: convergent key; deduplication; blockchain; differential authorization

ZHAO Tian[1], LI Hui[1], YANG Xin[1], WANG Han[1], ZENG Ming[2], GUO Haisheng[2], WANG Dezheng[2]

(1. Shenzhen Graduate School, Peking University, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202102009

http://kns.cnki.net/kcms/detail/34.1294.TN.20210519.1626.002.html, published online May 20, 2021

Manuscript received: 2021-01-13

Citation (IEEE Format): T. zhao, H. Li, C. Yang, et al., "Differentially authorized deduplication system based on blockchain," *ZTE Communications*, vol. 19, no. 2, pp. 67 - 76, Jun. 2021. doi: 10.12142/ZTECOM.202102009.

1 Introduction

In recent years, with the development of cloud storage technology, user data are uploaded to the cloud server and many copies of the data are repeatedly stored by different users, resulting in the waste of storage space. This makes the deduplication an urgent problem to be solved[1].

However, if the file data is directly stored in the cloud storage server, it faces a series of risks such as data theft. Therefore, we consider storing the ciphertext of the data in the cloud storage server.

In traditional encryption and decryption algorithms, the keys are generated independently by users leading to various ciphertexts of the same data, which makes the deletion of duplicate data difficult. If the cloud storage server generates the key and encrypts the data uniformly, the security of user data cannot be guaranteed once the cloud storage server is maliciously attacked and becomes untrustworthy.

In order to achieve deduplication under the premise of data security, a deduplication system based on convergent keys has been proposed in Ref. [2]. The encryption key $H(F)$ is obtained by hashing the data, and used to encrypt the user data. The convergent encryption makes the consistent ciphertext of the same file or data block, and the cloud storage server or external attackers cannot see the original data. It not only guarantees the confidentiality of the data, but also facilitates the cloud storage server to perform data deduplication by using the original data to generate a convergence key.

Because the encryption method of the convergent key is vulnerable to offline brute force cracking, semantic security cannot be guaranteed[3, 4]. In recent years, researchers in deduplication for convergent encryption have proposed a series of improvements. BELLARE et al.[5] proposed an information lock encryption scheme, which optimized key calculations and en-

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. 2019ZTE03-01, National Keystone R&D Program of China under Grant No. 2017YFB0803204, National Natural Science Foundation of China (NSFC) under Grant No. 61671001, Guangdong Provincial R&D Key Program under Grant No. 2019B010137001, Shenzhen Research Programs under Grant Nos. JCYJ20190808155607340, JSGG20170406144032901, JSGG20170824095858416 and JCYJ20170306092030521, and PCL Future Regional Network Facilities for Large-scale Experiments and Applications under Grant No. PCL2018KP001.

cryption methods. PUZIO et al.[6] designed the first repetition based on double-layer encryption in the deduplication scheme. The inner layer uses the convergent encryption schemes mentioned above, and the outer layer is outsourced to a trusted third party. In addition, BELLARE et al.[7] also described the DupLESS scheme, which adds an additional key to the convergent key generation process to invalidate the dictionary attack. LI et al.[8] proposed to use a deterministic secret sharing scheme instead of convergent encryption.

The above schemes are designed to alleviate the data security problem, but they do not fully consider how to build a credible authority when there are authority differences between users. Since the blockchain has the advantages of convenient generation and non-tampering, we consider introducing the blockchain technology to solve this problem.

NAKAMOTO Satoshi published the first paper on blockchain in 2008[9]. Although its main introduction focuses on Bitcoin, a digital currency payment system, blockchain has also caused extensive research in academia as its carrier. It allows any party that has reached an agreement to directly generate transactions without the participation of third-party intermediaries[10]. The blockchain encapsulates the history of consensus transactions in the block, as well as the identities of participants and timestamps. Each block uses the Hash algorithm to generate an important identification header for sequential connection, forming a chained data structure, which can be used as a distributed transaction and log record of the entire system[11].

Blockchain has the characteristics of decentralization, non-tampering, and traceability. Decentralization avoids the damage to the system caused by the evil master node in the traditional storage model; non-tampering ensures that if the attacker wants to tamper with a certain data in a block, he needs to recalculate the block and all the subsequent blocks; traceability guarantees that each user's operation can be located and tracked, which virtually increases its destruction cost.

The structure of the blockchain can be roughly divided into 6 levels, namely the data layer, network layer, consensus layer, incentive layer, contract layer and application layer. The data layer is at the bottom, which is mainly used to implement functions of data storage and transaction recording. The network layer is used to realize the functions of data transmission and verification. The consensus layer is the core part of the blockchain. It encapsulates various consensus algorithms. It is mainly used to achieve the consistency of block generation and transaction data. The incentive layer is mainly responsible for introducing incentive mechanisms, such as token distribution to miners who generate new blocks to encourage mining. The contract layer mainly includes various scripts written to enable the blockchain to obtain programmable application attributes. The application layer is the display of the blockchain in specific application scenarios, with many different manifestations. In this paper, we mainly use the first three layers of the blockchain. More specifically, the components of the consensus layer are utilized and improved.

According to different application scenarios, blockchains can be divided into the public blockchain, private blockchain and consortium blockchain. The public blockchain is completely open. Any user in the entire network is allowed to freely join or exit the blockchain system and everyone has equal rights. In the private blockchain, an organization has complete ownership of data on distributed nodes. The consortium blockchain is somewhere between the public and private blockchains, where several groups participate in the management of each node to grant different identities to jointly maintain the blockchain system.

We propose a differential authorization deduplication system based on blockchain, which alleviates the problems of single point of failure and inflexible permission changes. The user's public key and the permissions signed by the private key are written into the blockchain. It ensures the security of user permissions through the immutable modification of the blockchain and maintain each user's permission table. When the permission is changed, the blockchain directly generates a new block to cover the original permission, which is convenient for the dynamic modification of the permission. On the other hand, the blockchain can record each user's operation on the permission to ensure its traceability.

The rest of this paper is organized as follows. In Section 2, we describe the traditional differential authorization deduplication system and its problems. We also introduce the messaging process of the PoV algorithm. Then our differential authorization deduplication system based on blockchain is proposed in Section 3, followed by the performance analysis in Section 4 and experimental simulations in Section 5. Finally, conclusion and future works are drawn in Section 6.

2 Preliminaries

2.1 Traditional Structure

In this section, we introduce the traditional differential authorization deduplication system, including the main process of file upload and download, as well as its problems.

Let us consider a practical application scenario. In a company, subordinate relationships exist in different users leading to various permissions. For this reason, we have higher requirements for deduplication. Differential authorization deduplication should be implemented. Users with higher authority can upload and download data, while users with lower authority cannot download and access the data of high-level users.

The traditional differential authorization system[12] mainly provides different permission sets for different users. It introduces a private cloud server to maintain the permission table, and adopts the hybrid cloud architecture to realize the deduplication of differential authorization.

The system is mainly composed of three parts: the public storage cloud server provider (S-CSP) responsible for storing encrypted user data, the private cloud server responsible for maintaining the user permission table, and the user who uploads and downloads files.

The specific workflow is as follows:

1) System Initialization Stage

Define the tag of file F as $\varnothing_F = TagGen(F)$, and a label corresponds to a unique file data. Each permission p of the system has a corresponding permission key k_p. Define the token of file F as $\varnothing'_F = TagGen(F, k_p)$, that is, only users with permission p can access file F.

Assuming that the permission set owned by user U is P_U, its corresponding permission key $\{K_{P_i}\}, P_i \in P_U$ will be sent to the private cloud server acting as a permission check server. The private cloud server maintains a table to store each user's public key pk_U and its corresponding permissions.

2) File Upload

The file upload process is shown in Fig. 1.

Suppose that the file owner U wants to upload a file F for access by users with permission $\{P_j\}, P_j \in P_F$. First, the user needs to use his private key sk_U to verify his identity with the private cloud server. If the verification is passed, the user needs to send the tag $\varnothing_F = TagGen(F)$ of the file F to it. The private cloud server will return all initial file tokens $\{\varnothing'_{F,p} = TagGen(F,k_p)\}, p \in P_U$ that match the user's permissions, then the user will send these tokens to the S-CSP.

If duplicate files are found in S-CSP during upload, S-CSP first runs the Proof of Ownership (PoW) algorithm[8] to verify the user's ownership of the file. If it passes, it will return a file pointer to the user. A signature δ and a time stamp are appended to the token $\{\varnothing'_{F,p}\}$ and returned to the user. The user sends the token and the permission set $\{P_j\}$ of the file F to the private cloud server for verification. After the verification is passed, the private cloud server will calculate all the file tokens $\{\varnothing'_{F,p} = TagGen(F, k_p)\}, p \in P_F - P_U$ and return to S-CSP. The permissions of file F at this time are the union of the per-missions of P_F and other owners of the file.

If S-CSP does not find duplicate files when uploading, a signature δ and a time stamp are appended to the token $\{\varnothing'_{F,p}\}$ and returned to the user. The user sends the token to the private cloud server for verification. After passing the identity verification, the private cloud server will calculate all the file tags $\{\varnothing'_{F,p}\}$ within the P_F authority and return to the S-CSP, then the user can upload the data encrypted by the convergent key to the S-CSP.

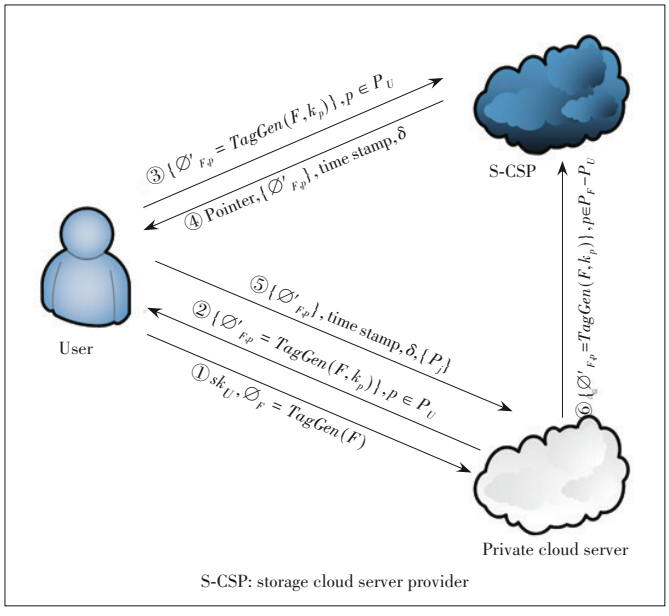
3) File Download

The user sends a file download request to the S-CSP, and the S-CSP will verify the user's permissions. If it cannot download, S-CSP will return the down-load failure. If it can download, S-CSP will return the encrypted data. The user uses the locally saved convergent key to decrypt the file and get the original data.

However, the data deduplication solution has some problems:

The first is the security assurance issue of the private cloud server. If the private cloud server is attacked and the user and corresponding permissions are tampered with, the system will not operate normally.

The second is the dynamic change of permissions. Once a file is uploaded, its permission is difficult to modify flexibly. When the user and file permissions change, for example, a user no longer has file permissions, the system cannot modify permissions in time.



▲**Figure 1. File upload**

2.2 Blockchain Consensus Algorithm

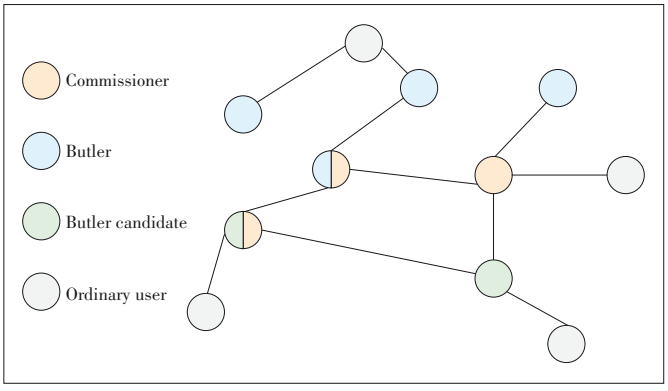
The design and implementation of blockchain involves many algorithms, the core of which is its consensus algorithm. For example, the consensus algorithm used by Bitcoin is the Proof of Work (PoW)[13]. The main idea of the PoW algorithm is that each independent node in the network conducts competitive mining to solve mathematical calculation problems, thereby obtaining the following accounting rights and generating new blocks. However, with the continuous mining of bitcoins, the mathematical puzzles that need to be solved to generate new bitcoins have become more and more complicated, which has caused a huge waste of computing resources and lower efficiency[14]. In addition, the important blockchain consensus algorithms include the Proof of Stake (PoS)[15, 16], Delegated Proof of Stake (DPoS)[17, 18], etc.

At present, there are many studies on the consensus algorithm of the blockchain. In this system, we use the blockchain based on the Proof of Vote (PoV) consensus[19] to construct the blockchain in the system. The PoV consensus can well ease double-spending attacks, selfish mining, witch attacks and

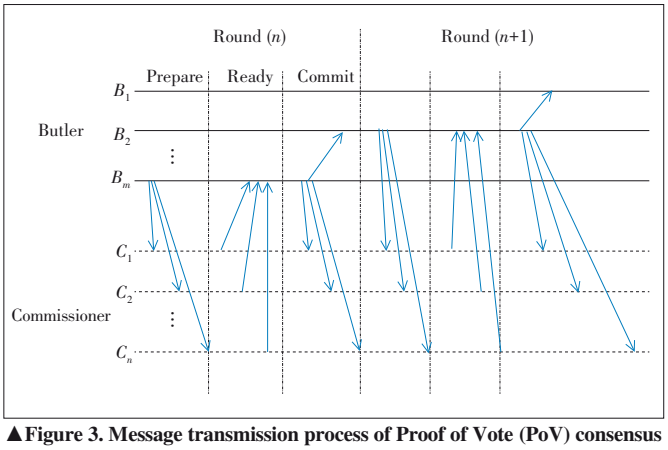
other attack methods. It also can well guarantee the security of user permissions. There are four types of nodes in this system, including commissioners responsible for voting, butlers responsible for accounting and production blocks, butler candidates, and ordinary user nodes that can apply to become butler candidates. This system allows concurrent roles to a certain extent, as shown in Fig. 2.

The consensus process of block generation is carried out jointly by a butler and all committee members. The butler is called duty butler, and the duty butler is determined by the butler number selected by the committee members. Assuming there are n commissioners, namely C_1, C_2,..., C_n; and m butlers, namely B_1, B_2,..., B_m, the consensus process of the block is shown in Fig. 3.

PoV divides a round of consensus into four phases: Prepare, Ready, Commit, Confirm. Among them, the Confirm stage is for the block placing, with no need to send messages. Each block contains a block header and an indefinite number of transactions. In the Prepare phase, the butler on duty takes a certain number of transactions from the transaction pool, packs them into pre-blocks and sends the pre-blocks to all committee nodes. The difference between the pre-block and the official block is that the pre-block does not have a timestamp, committee signature, and the number of the next butler on duty. The committee node needs to verify the block header of the received pre-block and the information contained in the transaction. If the verification passes, it will sign the pre-block header and send the signature to the duty butler.



▲Figure 2. PoV network model



▲Figure 3. Message transmission process of Proof of Vote (PoV) consensus

The duty butler can complete the pre-block and release the official block after collecting signatures of more than $n/2$ committee members. The node that receives the newly released block stores the block in the local blockchain and updates the relevant variables including the number of the duty butler, thereby replacing the duty butler who is responsible for the next round of consensus. The information supplement of the pre-block header depends on the signature of the committee member. The PoV stipulates that the latest member signature time of the signature is issued as the generation time of the block, and the next housekeeper number is generated by hashing the signature. These regulations make use of the randomness and unforgeability of signatures. Since the signatures generated by each committee node are random, the next butler number calculated is also random.

3 Differentially Authorized Deduplication System Based on Blockchain

In order to solve the above problems, this paper designs and implements a differentially authorized deduplication system based on blockchain. In more complex specific application scenarios, such as several companies working together to develop a project, they need to implement data deduplication on the same cloud storage server, which requires the system to credibly record the behavior of users to facilitate accountability. On the other hand, it is necessary to implement differential authorization of files according to different user identities and also to be able to make changes in time when users or file permissions change. Using the immutability and traceability of the blockchain to ensure the security of user permissions and accountability of behavior, these requirements can be met well. At the same time, when the user's file management authority changes, we can write the authority change into a new block. Because the blockchain is based on the record in the newly generated block, we can achieve dynamic changes in permissions.

The system is divided into three parts: the public cloud server S-CSP that stores encrypted data, the users who upload and download files, and the blockchain that saves permissions and upload and download records.

Blockchain is a distributed ledger and used in the Bitcoin currency transaction system. The blockchain network system maintains an orderly data block that keeps growing without a center. Each data block has a timestamp and a pointer to the previous block. Once the data is on the chain, it cannot be changed. Blockchain can be analogous to a distributed database technology. By maintaining a chain structure of data blocks, it can maintain a continuously growing, non-tamperable data record. The blockchain of our system is constructed

using the PoV algorithm described in the previous section, and its immutability is mainly guaranteed by the consensus mechanism.

As mentioned in Section 1, data deduplication achieved by convergent encryption can be performed in file-level data and block-level data respectively. In order to further save storage space and efficiently use bandwidth, we can encode the file into n data blocks $\{Bi\}$. When the files are not the same but the content is not much different, the data block deduplication check is used to complete the deduplication.

We will separately discuss the upload and download of file-level data and block-level data.

3.1 File-Level Data Upload and Download

1) System Initialization

Define the file F label as $\varnothing_F = TagGen(F)$, and a label corresponds to a unique file data. The user's permission set is $\partial = \{\partial_1 ... \partial_n\}$, where we define its number from small to large as the permission from high to low. Those with high-level permissions can access files uploaded by people with low-level permissions and modify file permissions. In the initial state of the system, the blockchain will have an authority table signed with private key S_{kp} of the negotiated highest authority owner P to declare the authority level of each user. Any legal user in the system can use its public key p_k to check the authority. At the same time, the blockchain will also store the label $\varnothing_F$ of the file F and the encrypted file permission ∂_F, which is convenient for S-CSP query. Suppose user U wants to upload a file with permission ∂_F, the user's private key is S_{kU} and S-CSP is initially blank.

2) File-Level Data Upload

The file-level data upload process is shown in Fig. 4.

The user sends the tag $\varnothing_F = TagGen(F)$ of the file to upload encryption by the private key S_{kU}, the user name U and its own authority ∂_U, and the S-CSP will query whether there is the tag $\varnothing_F = TagGen(F)$ of the file and the permission ∂_F of the file on the blockchain. If the file exists and the permission is lower than or equal to user permission ∂_U, the user needs to verify that he owns the file with S-CSP using the POW algorithm. At this time, the server will return a pointer to the user indicating that the server already has the file and the user has no need to upload repeatedly.

If the file does not exist or has no right to access the file, the user needs to write the uploaded relevant information to the blockchain, including the file tag $\varnothing_F$, the user name U, the user name U signed by the private key S_{kU} and the file at the access level $\{\varnothing_F, S_{kU}(U, \partial_F\}$. After verifying the identity of the user and S-CSP, the blockchain writes the record into a new block, and then the user can send the data of the file encrypted by the convergent key to the S-CSP.

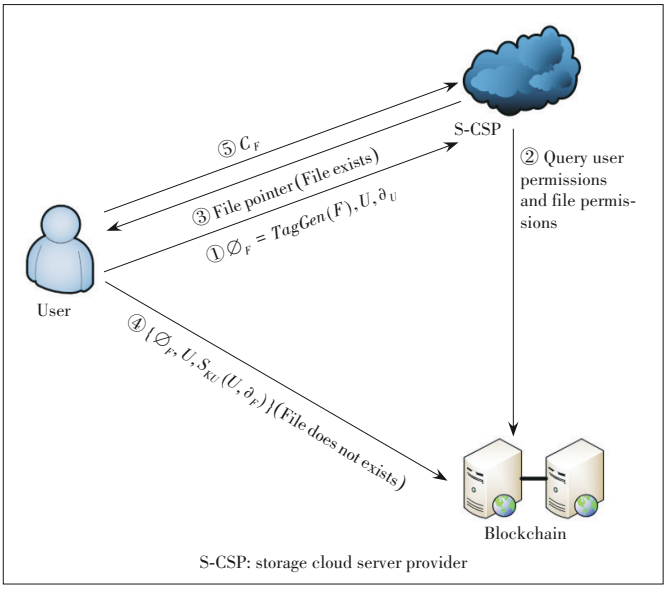
3) File-Level Data Download

The file-level data download process is shown in Fig. 5.

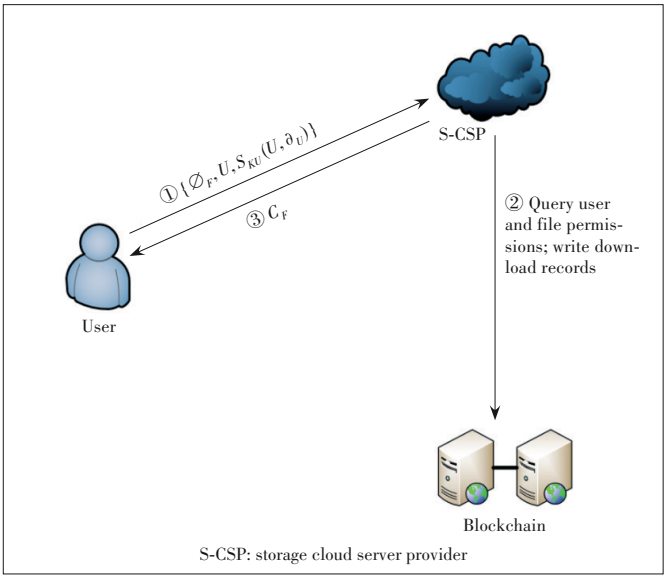
The user sends the tag $\varnothing_F = TagGen(F)$ of the file that will be downloaded to the S-CSP. The S-CSP will query whether the file exists. If the file does not exist, the S-CSP will return a prompt message to the user.

If the file exists, the user U sends the file tag $\varnothing_F$, the user name U, and the user name and authority information encrypted by its own private key S_{kU} :$\{\varnothing_F, U, S_{kU}(U,\partial_U)\}$ to S-CSP; S-CSP uses the user's public key P_{kU} to decrypt for obtaining authority, and then uses the public key of the highest authority to query the authority table to confirm whether the authority matches.

After passing, the S-CSP will send a confirmation message to the blockchain. After the blockchain verifies the identity of the user and S-CSP, the download record is saved in the block, and the S-CSP returns the file ciphertext C_F encrypted



▲Figure 4. File-level data upload



▲Figure 5. File-level data download

by the convergent key to the user. The user uses the locally stored convergent key to decrypt the file. LI et al.[20] have also done related research on the storage of convergent keys, but this is not the focus of this article. For simplicity, this article uses the traditional method of saving locally.

3.2 Block-Level Data Upload and Download

The block-level data upload and download process is similar to the file-level data upload and download process, as follows:

1) System Initialization

It is roughly the same as the system initialization requirements in Section 3.1, except that the file is divided into $\{Bi\}$ data blocks for upload and download.

2) Block-Level Data Upload

The user first sends the file F and all the tags $\varnothing_F$ and $\{\varnothing_{Bi}\}$ of the data block $\{Bi\}$, the user name U, and the own authority ∂_U to the S-CSP.

S-CSP will query whether there is a label for the file. If label $\varnothing_F$ of the file exists, it will turn to the processing flow of the file label in the previous section. The user can prove that he owns the file through PoW, and then S-CSP returns the corresponding pointer to inform the user that the file already exists. If the data block exists, the user needs to use the PoW algorithm to verify to the S-CSP that he owns the data block. At this time, the server will return a pointer to the user indicating that the data block already exists in the server, and there is no need to upload it repeatedly. S-CSP will add a new record to the blockchain, that is, the label of the newly added file corresponding to the data block, which is convenient for the repeatability check of the next file and data block.

If the data block does not exist, the user needs to write the uploaded relevant information to the blockchain, namely the file tag $\varnothing_F$, the data block tag $\varnothing_{Bi}$, the user name U, the user name signed by its own private key and the file can be accessed permission level: $\{\varnothing_F, \varnothing_{Bi}, \mathrm{U}, S_{kU}(U, \partial_F)\}$. After verifying the identity of the user and the S-CSP, the blockchain writes the information into the block, and then the user can send the data of the data block encrypted by the convergent key to the S-CSP.

3) Block-Level Data Download

The user sends the file label that will be downloaded to S-CSP. S-CSP will query whether there is a label for the file on the blockchain. If the file does not exist, S-CSP will return a prompt message to the user. If the file exists, the user sends the file label, data block label, user name and authority information encrypted by own private key: $\{\varnothing_F, \varnothing_{Bi}, S_{kU}(U, \partial_U)\}$ to S-CSP. S-CSP uses the user's private key to decrypt for obtaining the authority, and then uses the public key of the highest authority to query the authority table to confirm whether the authority matches. S-CSP will send a confirmation message to the blockchain; after passing the identity verification, the download record is saved in the block. The S-CSP returns the data block ciphertext encrypted by the convergent key to the user, and the user uses the locally stored convergent key to decrypt the data block. Then the user can restore the original data file $F=\{Bi\}$.

The encryption method of the convergent key has its inherent flaw, that is, it cannot resist offline dictionary attacks. Specifically, if external attackers know the ciphertext C_F and can infer the file set $\{F\}$, they can directly generate the convergent key and encrypt the corresponding files for comparison. If the ciphertext is the same, attackers may obtain the original data file.

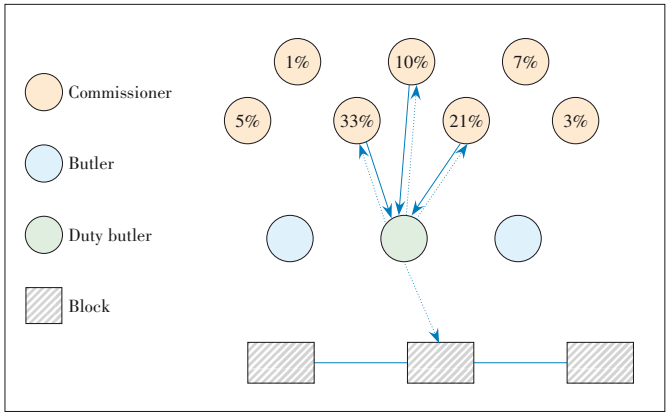
In response to this defect, this article proposes to use block-level data upload when storing high-privilege files, and use double-layer encryption for more important data blocks (in the application scenario of deduplication, the number of important data blocks in the file is less), that is, use another *Hash* function to generate the convergent key in the outer layer of the encrypted data block to encrypt again. After S-CSP receives the data, it sends a message to all users higher than the authority, calculates the convergent key of the *Hash* function and saves it. When the user needs to download a file, the convergent key is used to decrypt the outer layer, and then the original convergent key is used to decrypt the inner layer. This mode can be turned on or off according to the requirements of the system application scenario.

3.3 Specific Application of PoV Algorithm in The System

This section mainly introduces the specific consensus and generation process of the blockchain used to store user authority information. In this system, we introduce the PoV consensus algorithm in Section 2.2 to generate our blockchain. The consensus process is roughly the same as the foregoing. Specifically, we can implement the PoV algorithm in the initial state with the company's leadership as the committee node and ordinary employees as the ordinary nodes. The ordinary employees can submit a campaign request, and the members of the leadership will vote to select trusted butler nodes and the current duty butler nodes. The initial block maintains the permissions of all initial users. After the initial block is confirmed by all members, the selected butler node on duty is responsible for generating a new block to record the permissions of all users and the upload and download records of files. The specific block generation process is shown in Fig. 6. The percentage in the figure indicates the weight of commissioners.

Different from the traditional PoV scheme, we no longer require the replacement of the duty butler after a specific block is generated. Instead, the committee members vote to elect a new butler on duty. This is because in actual application scenarios, a housekeeper on duty is probably not online. At this time, if you need to upload or download files or change permissions, committee members need to select a new butler on duty to handle the request.

In addition, we have also changed the voting weight of the



▲Figure 6. Block generation process

committee members, that is, the voting weight of each committee member is no longer equal, but the voting weight of each committee member is different according to their different rights. This improvement has brought the following benefits: First, it is in line with the logic of inter-company affairs and people with higher status can have more voice to determine file access and permission changes; second, when there is an emergency, it only needs to notify a few committee nodes to make the voting weight more than half to complete the fast processing of the transaction; finally, when the user scale is expanded to a large scale, the voting scheme can speed up the consensus completion time and improve the transaction processing effectiveness.

4 Theoretical Analysis

In this section, we analyze the performance of the system, including functional analysis and safety analysis.

4.1 Functional Analysis

4.1.1 Differential Access Control

The main process of uploading and downloading data of the system has been introduced. The specific differential authorization is embodied in the user uploading a file, which can be easily written into the file's permissions, including reducing or increasing file permissions. For example, a fourth-level permission can upload a fifth-level permission file for low-privilege access; it can also upload a high-level file such as the second-level authority. Users with a level greater than or equal to the second-level authority can access the file, and at the same time, the third-level user has no right to access the file, which realizes the system's differential authority access control.

4.1.2 Dynamic Changes in Permissions

The system can also solve the problem of dynamic changes in permissions.

When the user authority changes, we can update the record in the block, generating a new block record. According to the traceability of the blockchain, when all nodes confirm the block, the user will have the new authority. At this time, if the authority level is increased, the user can access and download the high-level authority file. If the authority is reduced, the user cannot access the original authority file.

When the file authority changes, the high-level authority or the file uploader can send information to the blockchain again, rewrite the file authority level, and realize the dynamic management of file authority.

4.1.3 Single-Enterprise Environment

Although the system mainly considers the differential authorization deduplication between multiple institutions, it is also applicable to a single-enterprise environment. Using the typical structure of blockchain, the generation of new blocks is relatively simple. At the same time, the PoV consensus mechanism has strong consistency, which means that when the user or file permissions change, the change can be quickly confirmed, thus ensuring the efficiency of system operation.

As mentioned earlier, in the context of multi-institution, using this scheme can ensure the security of authority management. If you use this solution internally, you can deploy the blockchain on the cloud. Nodes located in different data centers act as butler nodes, so that when a butler node fails, the entire blockchain can still operate stably, which greatly improves the availability of the system.

4.2 Performance Analysis

Compared with the traditional scheme, the additional cost of this system mainly includes the following aspects, namely, the time for generating a new block by consensus, the time for permission query, and the time required for outer encryption.

For simplicity, we use some symbols to represent these variables, as shown in Table 1.

It can be seen that in our solution, the time required to complete the upload and download of the entire file is:

$T_A = T_{GB} + T_{PQ} + T_{FL} + T_{CE} + T_{FT}$.

In the traditional solution, because the permission table is maintained on the private cloud server, the permission query speed is very low and can be ignored. At this time, the total time required by the traditional solution is:

$T = T_{FT} + T_{FL} + T_{CE}$.

▼Table 1. Parameters and their description

Parameter	Description
T_{FL}	Time required for file label generation
T_{CE}	Time it takes for the file or data block to convergent encryption
T_{FT}	File transfer time
T_{PQ}	Time required for blockchain permission query
T_{OE}	Time required for outer encryption
T_{GB}	Time required to generate a block
T_A	Total time
T_{AE}	Enhancement plan total time

We set $\omega_1 = (T_{GB} + T_{PQ})/T$. At this point, if ω_1 is small enough, it proves that our solution achieves dynamic management of permissions without introducing obvious additional overhead.

In the enhancement scheme, the time we need to complete the entire process is $T_{TE} = T_A + T_{OE}$.

We set $\omega_2 = T_{OE}/T_A$. It can be seen that if ω_2 is small enough, that is the number of blocks that need to be double-layered encryption is relatively small, or the time used for outer encryption is relatively small, it proves that we can ensure the security of the system without significantly increasing the time overhead.

4.3 Security Analysis

The security of permissions is mainly guaranteed by the security of the blockchain, that is, the tolerance of attacking nodes.

Define the number of blockchain nodes as n and the number of attackers as f, when the weight of each node is the same, our tolerance for attacking blocks is:

$$f \leqslant \frac{n-1}{2}.$$

That is, it can tolerate no more than half of the nodes being attacked, which better guarantees that the permissions cannot be tampered with.

At the same time, the double-layer encryption scheme for confidential data blocks mentioned in Section 3 can also better prevent offline dictionary attacks. The encryption method we use for convergent encryption is Advanced Encryption Standard (AES) encryption, and the key length is 256 bit. AES encryption has good resistance to brute force cracking. If 10 000 collision attacks are executed every nanosecond, it will take 1.8×10^{56} years to crack[21].

5 Experimental Simulation

We implemented the model of the system and ran it in our own experimental environment. The code was implemented in C++. Table 2 shows the hardware conditions for the implementation environment.

We tested the performance of this model. The main measurement indicators are the ratio of the time used for permission query, file transfer, file label generation, and convergent encryption when the upload and download file sizes are different.

We used AES encryption as the encryption algorithm for convergent encryption. The key is a 256-bit hash value generated by the SHA-256 algorithm; the file label is generated by the SHA-1 algorithm. When the permission changes, we can manually set the new block generation time to 1 s, which is the time required for the permission change.

▼Table 2. Hardware conditions for the implementation environment

Hardware	Setting
Operating system	Ubuntu16.04
CPU brand	AMD
CPU frequency	1.7 GHz
Memory size	8 GB
Network bandwidth	1 000 Mbit/s

We tested the performance of the system when the file size was 100 MB, 200 MB, 300 MB, and 400 MB (Fig. 7).

Because the authorization query time is too short, the figure shows the total time used for 10 000 queries. It can be seen from the experimental results that compared to data transmission and convergent encryption, the time required for user permission query and record writing to the blockchain is shorter. Our system does not significantly increase system overhead while achieving differential authorization deduplication.

At the same time, we also tested the enhancement scheme proposed in Section 3. The main performance indicators are the ratios of the time used for double-layer encryption of important data blocks, file label generation, convergent encryption and file transfer.

In this enhanced scheme, the outer layer encryption uses the AES encryption algorithm, and the key is a 128-bit hash value generated by the MD5 hash algorithm. The inner layer encryption uses the AES encryption algorithm, and the key is a 256-bit hash value generated by the SHA-256 algorithm.

We tested the situation where the number of important data blocks (requiring double-layer encryption) accounted for different proportions of the total number of data blocks. For convenience, we set the number of important data blocks to 1, the size of 100 MB, and the size of ordinary data blocks to 100 MB. The experimental results are shown in Fig. 8.

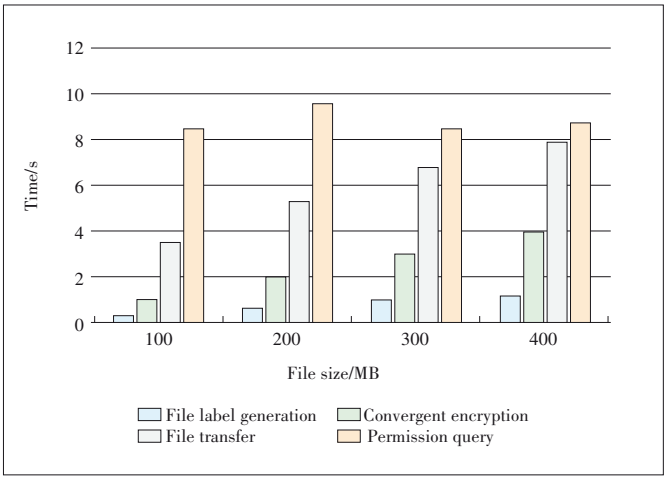
It can be seen from the experimental results that when the important data is relatively few, our enhancement scheme will not significantly increase system overhead while improving data security.

In summary, from the experimental results, the maximum value of ω_1 is less than 1%, and the maximum value of ω_2 is less than 5%, indicating that our solution does not significantly increase the overhead.

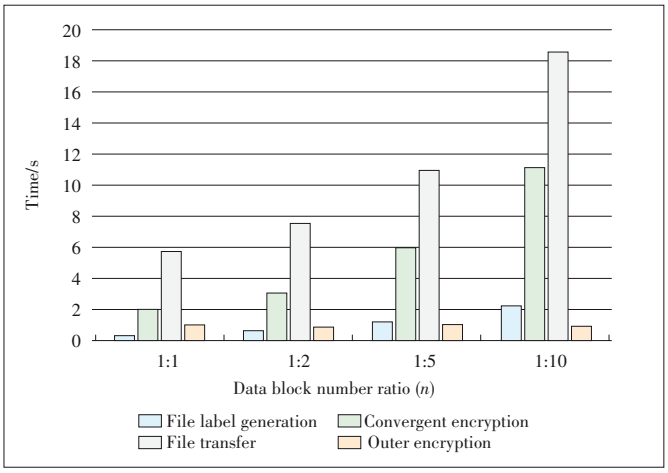
6 Conclusions and Future Work

The differentially authorized deduplication system based on blockchain system proposed in this paper writes the user permission table and file permissions into the blockchain, using the immutable modification of the blockchain, and it better solves the vulnerability of public cloud servers and private cloud servers. It can also overcome the single point of failure problem caused by the original private cloud server due to centralization.

At the same time, the blockchain read and write efficiency is high, and the latest written information will prevail. Therefore, it can solve the problem that the permissions of users and



▲**Figure 7. Permission query time and main process time**



▲**Figure 8. Time required for outer encryption and the main process time**

files in the original system cannot be dynamically modified in time, and realize the management of dynamic changes in user and file permissions. At the same time, the experimental results show that the extra cost of our proposed scheme and enhancement scheme is less than 5%, indicating that the system overhead is not significantly increased.

In the future work, we will continue to improve the existing convergent key generation method or explore other encryption schemes to replace the existing convergent key encryption to overcome its vulnerability to offline dictionary attacks.

References

[1] HARNIK D, PINKAS B, SHULMAN-PELEG A. Side channels in cloud services: Deduplication in cloud storage [J]. IEEE security & privacy, 2010, 8 (6): 40 – 47. DOI: 10.1109/MSP.2010.187
[2] DOUCEUR J R, ADYA A, BOLOSKY W J, et al. Reclaiming space from duplicate files in a serverless distributed file system [C]//22nd International Conference on Distributed Computing Systems. Vienna, Austria: IEEE, 2002: 617 – 624. DOI: 10.1109/ICDCS.2002.1022312
[3] LIU J, ASOKAN N, PINKAS B. Secure deduplication of encrypted data without additional independent servers [C]//22nd ACM SIGSAC Conference on Computer and Communications Security. Denver, USA: ACM, 2015: 874 – 885. DOI: 10.1145/2810103.2813623
[4] LIU X F, SUN W H, LOU W J, et al. One-tag checker: message-locked integrity auditing on encrypted cloud deduplication storage [C]//IEEE Conference on Computer Communications (INFOCOM). Atlanta, USA: IEEE, 2017: 1 – 9. DOI: 10.1109/INFOCOM.2017.8056999
[5] BELLARE M, KEELVEEDHI S, RISTENPART T. Message-locked encryption and secure deduplication [C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques: Berlin/Heidelberg, Germany: Springer, 2013: 296 – 312. DOI: 10.1007/978-3-642-38348-9_18
[6] PUZIO P, MOLVA R, ÖNEN M, et al. ClouDedup: secure deduplication with encrypted data for cloud storage [C]//5th International Conference on Cloud Computing Technology and Science. Bristol, UK: IEEE, 2013: 363 – 370. DOI: 10.1109/CloudCom.2013.54
[7] KEELVEEDHI S, BELLARE M, RISTENPART T. Dupless: server-aided encryption for deduplicated storage [C]//22nd USENIX Conference on Security. Washington D. C., USA: USENIX, 2013: 179 – 194
[8] LI J, CHEN X F, HUANG X Y, et al. Secure distributed deduplication systems with improved reliability [J]. IEEE transactions on computers, 2015, 64(12): 3569 – 3579. DOI: 10.1109/TC.2015.2401017
[9] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system [EB/OL]. (2008-10-31)[2020-01-01]. https://bitcoin.org/bitcoin.pdf
[10] KHALIL R, GERVAIS A. Revive: rebalancing off-blockchain payment networks [C]//ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA: ACM, 2017: 439 – 453. DOI: 10.1145/3133956.3134033
[11] BHATTACHARYA R, WHITE M, BELOFF N. A blockchain based peer-to-peer framework for exchanging leftover foreign currency [C]//Computing Conference. London, UK: IEEE, 2017: 1431 – 1435. DOI: 10.1109/SAI.2017.8252284
[12] HALEVI S, HARNIK D, PINKAS B, et al. Proofs of ownership in remote storage systems [C]//18th ACM Conference on Computer and Communications Security. Chicago, USA: ACM, 2011: 491 – 500. DOI: 10.1145/2046707.2046765
[13] GERVAIS A, KARAME G O, WÜST K, et al. On the security and performance of proof of work blockchains [C]//ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria: ACM, 2016: 3 – 16. DOI: 10.1145/2976749.2978341
[14] EYAL I, SIRER E G. Majority is not enough: Bitcoin mining is vulnerable [C]//International Conference on Financial Cryptography and Data Security. Berlin/Heidelberg, Germany: Springer, 2014: 436 – 454. DOI: 10.1007/978-3-662-45472-5_28
[15] KIAYIAS A, RUSSELL A, DAVID B, et al. Ouroboros: a provably secure proof-of-stake blockchain protocol [C]//Advances in Cryptology—CRYPTO 2017, Cham, Switzerland: Springer, 2017: 357 – 388. DOI: 10.1007/978-3-319-63688-7_12
[16] LI W T, ANDREINA S, BOHLI J M, et al. Securing proof-of-stake blockchain protocols [C]//Data privacy management, cryptocurrencies and blockchain technology. Cham, Switzerland: Springer, 2017: 297 – 315. DOI: 10.1007/978-3-319-67816-0_17
[17] ZHENG Z B, XIE S A, DAI H N, et al. An overview of blockchain technology: Architecture, consensus, and future trends [C]//International Congress on Big Data (BigData Congress). Honolulu, USA: IEEE, 2017: 557 – 564. DOI: 10.1109/BigDataCongress.2017.85
[18] SANKAR L S, SINDHU M, SETHUMADHAVAN M. Survey of consensus protocols on blockchain applications [C]//4th International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore, India: IEEE, 2017: 1 – 5. DOI: 10.1109/ICACCS.2017.8014672
[19] LI K J, LI H, WANG H, et al. PoV: an efficient voting-based consensus algorithm for consortium blockchains [J]. Frontiers in blockchain, 2020, 3: 11 DOI: 10.3389/fbloc.2020.00011

[20] LI J, CHEN X F, LI M Q, et al. Secure deduplication with efficient and reliable convergent key management [J]. IEEE transactions on parallel and distributed systems, 2014, 25(6): 1615 – 1625. DOI: 10.1109/TPDS.2013.284
[21] KAHATE A. Cryptography and network security [M]. New Delhi, India: Tata McGraw-Hill Education, 2013

Biographies

ZHAO Tian is a postgraduate of the Shenzhen Graduate School, Peking University, China. His main research directions are big data applications, blockchain, and distributed storage.

LI Hui (huilihuge@163.com) is currently a professor and Ph.D. supervisor with the Shenzhen Graduate School, Peking University, China. His research fields include cyberspace security and block-chain technology, artificial intelligence and future network systems, distributed storage coding theory and systems, intelligent big data analysis, and data standards.

YANG Xin received the B.Eng. degree from the Department of Computer Science and Engineering, South China University of Technology, China in 2016. She is currently pursuing the Ph.D. degree with the School of Information Science, Peking University, China. She is also the student of the Peng Cheng Laboratory, China. Her research interests include cyber security, future network architecture, and distributed storage systems.

WANG Han received the B.Eng. degree from the Department of Communication Engineering, Jilin University of Technology, China in 2017. She is currently pursuing the Ph.D. degree with the School of Information Science, Peking University, China. Her research interests include distributed systems and cyber security.

ZENG Ming received his bachelor's degree from University of Electronic Science and technology, China, majoring in computer science and technology. He is a system architect of ZTE Corporation and has been engaged in software development and architecture design for more than 16 years. His research interests include big data, blockchain and other related technologies fields.

GUO Haisheng received his master's degree from Nanjing University of Aeronautics and Astronautics, China. He is a project manager of ZTE Corporation and has been engaged in software development, architecture design, project management for nearly 20 years. His research interests include blockchain and other related technologies fields.

WANG Dezheng received his master's degree in computer science from Zhejiang University, China. He is a chief engineer of the Center Institute, ZTE Corporation and has been engaged in architecture design for more than 20 years. His research interests include big data and blockchain.