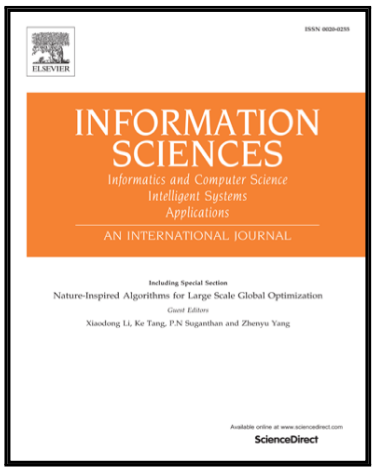

Block-Secure: Blockchain Based Scheme for Secure P2P Cloud Storage

Jiaxing Li, Jigang Wu, Long Chen

PII: S0020-0255(18)30501-2
DOI: 10.1016/j.ins.2018.06.071
Reference: INS 13771

To appear in: *Information Sciences*

Received date: 1 March 2018
Revised date: 21 June 2018
Accepted date: 23 June 2018

Please cite this article as: Jiaxing Li, Jigang Wu, Long Chen, Block-Secure: Blockchain Based Scheme for Secure P2P Cloud Storage, *Information Sciences* (2018), doi: 10.1016/j.ins.2018.06.071

Block-Secure: Blockchain Based Scheme for Secure P2P Cloud Storage

Jiaxing Li[a], Jigang Wu[a,b,*], Long Chen[a]

[a] *School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China*
[b] *Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China*

Abstract

With the development of Internet technology, the volume of data is increasing tremendously. To tackle with large-scale data, more and more applications choose to enlarge the storage capacity of users' terminals with the help of cloud platforms. Before storing data to an untrusted cloud server, some measures should be adopted to guarantee the data security. However, the communication overhead will increase dramatically when users transmit files encrypted by a traditional encryption scheme. In this paper, we address the above problems by proposing a blockchain-based security architecture for distributed cloud storage, where users can divide their own files into encrypted data chunks, and upload those data chunks randomly into the P2P network nodes that provide free storage capacity. We customize a genetic algorithm to solve the file block replica placement problem between multiple users and multiple data centers in the distributed cloud storage environment. Numerical results show that the proposed architecture outperforms the traditional cloud storage architectures in terms of file security and network transmission delay. On average, the file loss rate based on the simulation assumptions utilized in this paper is close to 0% on our architecture while it's nearly 100% and 71.66% on the architecture with single data center and the distributed architecture using genetic algorithm.

*Corresponding author
Email address: asjgwucn@outlook.com (Jigang Wu)

Besides, with proposed scheme, the transmission delay on the proposed architecture is reduced by 39.28% and 76.47% on average on the user's number and the number of file block replicas, respectively, in comparison to the architecture with single data center. Meanwhile, the transmission delay of file block replicas is also reduced by 41.36% on average than that on the distributed architecture using genetic algorithm.

Keywords: Cloud storage, Security, Blockchain, Architecture, Distributed

1. Introduction

Cloud storage is a kind of system with distributed data centers that takes advantage of virtualization technology and provides interfaces for data storage. It also makes servers or data centers be able to work together for conveniently sharing and accessing resources. Recently, cloud storage has received massive attractions in personal and business organizations because it's convenient and efficient.

In order to access application resources from anywhere and at any time, users have to move their data into the cloud. Cloud provides benefits like flexibility, automatic software update, disaster tolerance, cost reduction and etc. For advantages, challenges and key technologies in different types of cloud storage one can refer to [40, 13].

It's important to protect users' privacy [46] and data security [47] because the data may leak when users store their data in the cloud. Meanwhile, the influence of cloud storage security is widely and there is an increasing public concern about users' privacy [36, 35]. However, there still exists weakness for the existing cloud storage architectures, such as centralized data storage that severely harms server physical security and the need for trusted third-party which are nightmares for the privacy of users' data.

For the current distributed cloud storage, the data stored in several data centers are not fully distributed. The data are still stored in several data centers at high density, and a massive amount of data will be leaked even if one of

the data centers was broken down. Public media all around the world have repeatedly reported security issues related to cloud storage in recent years, such as users' privacy file leakage on iCloud [25]. Unfortunately, there are still no effective solutions for the security of distributed cloud storage.

Zyskind et al. propose an architecture and their architecture uses blockchain to protect personal data through distributed storing file access permissions in the blockchain, but its data storage still uses a centralized cloud and requires a trusted third-party to support [50]. Blockchain is a distributed database system, it also can be regarded as a number of nodes jointly maintained by the Distributed Ledger Technology (DLT), which is difficult to tamper, forge and trace [29]. The blockchain records all the information of the transactions, and once the data enter the blockchain, almost nobody can change it. This unchangeable feature is not derived from the use of a certain operation but from the blockchain system and the mechanism itself. This makes the use of blockchain technology easier and more secure than other security technologies. For example, the work [26] introduces how to use blockchain technology in intrusion detection systems (IDSs) and [19] utilizes blockchain to protect the security of users' data.

As its typical feature of blockchain, which includes peer-to-peer (P2P) communication, we utilize unstructured P2P network in our distributed storage architecture. A P2P network shares resources among nodes rather than concentrating them in a single data center or server [31]. In a P2P system, nodes are pooled together to provide their network resources [34]. Therefore, we can put all users' vacant storage space as a storage pool providing a cloud storage service to other users.

In this new cloud storage architecture, we need to solve the following research challenges. i) A novel storage strategy is necessary because existing distributed cloud storage strategy stores data centrally in several data centers and once a data center is broken down, a large amount of data may leak. ii) In order to compare the network performance, a heuristic optimization algorithm should be utilized to optimize the NP-hard problem of the files transmission

delay between users and data centers. iii) Storing file blocks randomly into the nearest nodes in the network that have vacant storage capacity is an intuitive challenge because this process will consume a massive amount of computation and communication resources. Especially in the large-scale network, where the computational complexity will grow exponentially.

To deal with the disadvantages and challenges as mentioned above, we integrate distributed cloud storage and blockchain technology to propose a blockchain-based distributed cloud storage architecture that can provide secure and reliable cloud storage services for enterprises or individual users. In this work, related works on cloud storage are first to introduce, then a blockchain-based distributed cloud storage architecture and its security analysis are proposed and finally, three cloud storage architectures including a centralized cloud storage architecture and a distributed cloud storage architecture are compared through simulations. To the best of our knowledge, this paper is the first work for the blockchain-based architecture with completely distributed cloud storage.

This paper is distinct from our previous work [12] in many aspects. First, [12] has been significantly extended by adding investigations for the further security analysis and the analysis of replica number in all three architectures. Moreover, the numerical results of further security analysis and network performance analysis on replica number are provided in this paper. The main contributions of this paper are summarized as follows.

- We propose a blockchain-based distributed cloud storage architecture to provide more secure and reliable cloud storage services for enterprises or individual users.
- We customize a genetic algorithm to solve the file block replica placement problem between multiple users and multiple data centers in distributed cloud storage environment.
- We conduct simulations on the proposed architecture for security and network performance, in comparison with other two different cloud storage architectures.

- On average, the file loss rate in this work is almost 0% based on the assumptions made in the simulation for our architecture while it's nearly 100% and 71.66% for the other two architectures.

- The transmission delay on the proposed architecture is averagely reduced by 39.28% and 76.47% on the user's number and the number of file block replicas, respectively, in comparison to the architecture with single data center.

The rest of this paper is organized as follows. In Section 2, we discuss related work of cloud storage and blockchain technology. Then we propose a novel blockchain based distributed cloud storage architecture in Section 3. The network performance and security analysis are presented both for the proposed architecture and for the traditional cloud storage architecture in Section 4. Numerical simulations are presented in Section 5. We conclude this paper in Section 6.

2. Related Works

Cloud storage is a kind of Internet technology for sharing resources with IT-related capabilities and it is important to either enterprises or individual users. Traditional security strategies mainly focus on information encryption [21][45][18], data deduplication [16, 15, 14], access control [37, 33, 1, 13], privacy-preserving keyword search [23, 39, 10, 17], network performance improvement [22] and etc. Recently, application data are becoming more and more intensive and a separate cloud cannot meet the storage demands of users. To deal with the situation mentioned above, the Software Defined Storage (SDS) integrates a number of distributed cloud storage services [38]. When a cloud cannot meet the demands of users, their requests can be transferred to other cloud platforms.

Compared with traditional cloud storage, the heterogeneity among cloud service providers such as different device types, hardware composition and etc. can be properly handled by SDS. This difference can be shielded by software-defined

hardware [49] and software decoupling methods [5] to provide technical support for the aggregation of the upper storage resources and the unified scheduling platform. Inspired by SDS, the storage strategy in our architecture is a random storage strategy which takes users' fixed vacant storage space as the cloud storage space and then rents it to other users who need storage space. From the cloud service provider's perspective, the marginal cost of cloud resources is also increasingly prominent because of the demands to maintain a large number of servers and services. From another point of view, if we put the users' vacant storage space as cloud storage space, the cloud storage infrastructure costs will be greatly reduced. However, there are still many critical security issues in cloud storage.

In public-key systems, public-key cryptography (PKC) plays a significant role in information security using a public-private key pair in which one is for encryption and the other is for decryption [3]. The private key is kept secret and the public key is released to public. Besides, the public keys should be associated with the users in a trusted or authenticated way by a public key infrastructure (PKI). It is well known that certificate based cryptography systems are most widely deployed public-key cryptography systems. However, these certificates have to be generated in large and distributed to many users. Moreover, frequent verification is required for the certificates. Thus, the management of public-key certificates is complicated. In order to avoid the disadvantages of using public key certificates, Shamir et. al [32] introduced the concept of identity-based cryptography. Identity-based cryptography means that public keys can be generated directly from user identifiers, such as email addresses, telephone numbers, bank account number and etc.

In the aspect of storage security, authors of [50] have proposed a blockchain based solution for cloud storage. However, it only considers individual secure cloud storage rather than the security of the whole system. Blockchain is a bottom technology behind Bitcoin proposed by Nakamoto [29]. Just like its name, a blockchain is composed of blocks and a chain, in which blocks verify their own former and latter blocks mutually through a hash chain. It mainly

combines hash operation, P2P network, digital signature and Merkle Hash Tree to provide a secure, open, transparent, consensus and reliable service.

Before deploying blockchain technology in cloud storage, it must satisfy a condition that honest nodes constitute of at least half the computational power in the network [8]. In blockchain, transactions are stored in the block and chain is supported by a hash operation like SHA-256 [6]. The Bitcoin's blockchain protocol randomly chooses one processor (miner) per 10 minutes which issues a proposal that everyone adopts to reach an agreement by broadcasting the proposal [29]. Meanwhile, the authors of [41] have also proposed a blockchain-based P2P cloud storage network named *Storj* which implements end-to-end encryption allowing users to transfer and share data without a reliance on a third party data provider. Unlike [41], this paper focuses on the design of blockchain-based architecture for distributed cloud storage, where both the security of users data and the security of the architecture are considered. Moreover, we have also provided analysis for scheme performance in security and in network delay, together with the comparison with existing two schemes, those are not presented either in [41]. On the cloud storage architecture, prior works (e.t. the two schemes) mainly proposed two cloud storage architectures, i.e., multiple users with single data center and multiple users with multiple data centers [44].

Optimizing distributed cloud storage service transmission time resembles the resource scheduling optimization problem, and authors in [2] use a genetic algorithm to optimize the data resource scheduling between the scientific application and task of users' requirements. Besides, in cloud storage architecture, replication is one of the significant data reliability techniques [28]. Therefore, we can utilize the genetic algorithm to optimize our distributed cloud storage replicas transmission time. Furthermore, in order to improve cloud storage security in architecture, we combine the distributed cloud storage architecture with blockchain technology.

As mentioned above, the works on cloud storage security cannot be directly extended to solve the blockchain-based secure storage problem without a third party. However, some earlier works are also rarely considering on an architecture

level, for examples, authors of [4] propose a blockchain-based system with private keyword search for secure data storage, authors of [20, 43] respectively propose a blockchain-based data integrity checking framework and remote checking scheme for cloud storage, and authors of [42] propose a blockchain-based publicly verifiable data deletion scheme for cloud storage. Moreover, the security of cloud storage architecture is not the simple overlay of multiple specific security technologies. Thus, this paper studies a new blockchain-based security architecture design for distributed cloud storage to improve the security of the distributed cloud storage system.

3. Architecture Design

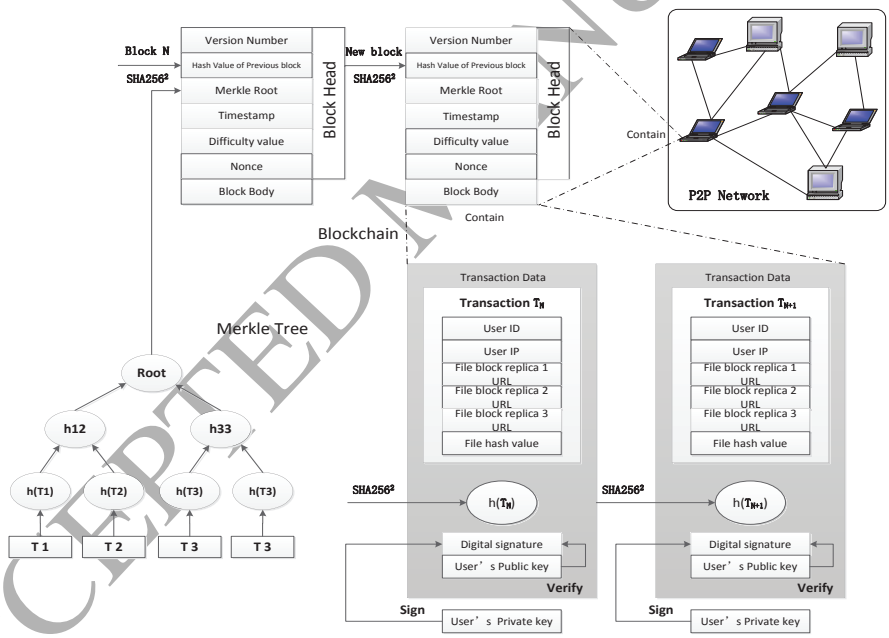

Figure 1: Distributed Cloud Storage Architecture Based on Blockchain

In this section, we present a blockchain-based security architecture for distributed cloud storage. As can be seen in Fig. 1, in this architecture, we first divide users' files into several blocks with the same size, encrypt these file blocks,

sign them through a Digital Signature Algorithm (DSA) and upload them to a P2P network. Then we utilize blockchain technology as a trading mechanism between users who need cloud storage service and users who supply their vacant storage space. Furthermore, we choose a random file replica placement strategy in this architecture so that users can retrieve their files quickly from the cloud and alleviate the burden of the P2P network. Finally, file integrity verification will be ensured by using the Merkle Hash Tree as a validation method.

3.1. Architecture Overview

3.1.1. Files are chunked, encrypted and uploaded to P2P network

Considering the network performance, users' files need to be chunked and encrypted before they are uploaded to the P2P network. Actually, almost all users' files are split into blocks of the same size, which is limited by network protocol and is convenient for transmitting data packages. For security, files should be encrypted before they are uploaded to the cloud so that users' information will not be retrieved. In the proposed architecture, users run an elliptic curve cryptography (ECC) based key generation algorithm that uses the secp256k1 curve to generate a public-private key pair (pk,sk) to encrypt and decrypt their files without any key generation center (KGC) or third party. Besides, a signature key pair (spk,ssk) will also be generated by an ECC-based digital signature algorithm named ECDSA.

3.1.2. Use blockchain as a trading mechanism

A blockchain storing a time-ordered collection of widely accepted transactions, is an append-only distributed database [24]. The blockchain technology enters peoples' sight after the success of Bitcoin proposed by Nakamoto in [29]. In Bitcoin, transaction size attracts many concerns and files cannot be stored in the blockchain directly.

As shown in Fig. 2, in the proposed architecture we can store file hashes, file location URLs (Uniform Resource Locator), file replicas location URLs and etc. instead of file blocks themselves in the blockchain. It's noticeable that each user

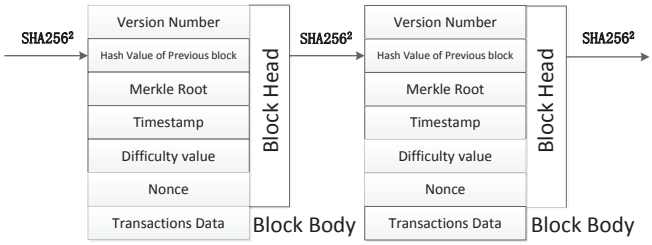

Figure 2: Structure of Blockchain

has a copy of all transactions in the blockchain and the size of the transaction information is negligible to the user's hard disk so that the architecture can reduce a massive amount of memory space for users. In this architecture, an adversary cannot get anything about the raw users' file data from the blockchain, as only URLs and hash values are stored in it.

3.1.3. File storage strategy and file replicas replacement

Compared with traditional cloud storage architecture, our distributed cloud storage architecture stores file blocks to nodes in a P2P network randomly. Because a fault tolerance mechanism is necessary for every intelligent system, our architecture achieves the fault tolerance mechanism by using file replicas as data redundancy. File replicas will be stored in the P2P network randomly, and their URLs will be stored in blockchain after being encrypted so that users can know and get their own file completely. The number of file replicas is determined by the network performance influencing by the file replicas placement strategy and the number of file blocks replicas.

3.1.4. File integrity verification

As shown in Fig. 3, Merkle Hash Tree (MHT) is constructed by calculation results based a one-way cryptographic hash operation like SHA256 [11]. Besides, $SHA256^2$ is two times SHA256 encryptions. A MHT is first constructed by pairing data (e.g. in the Bitcoin system it usually refers to transactions), next hashing the pairs, then pairing and hashing the results until a single hash remains, the Merkle Root. In the tree, each leaf node containing information can

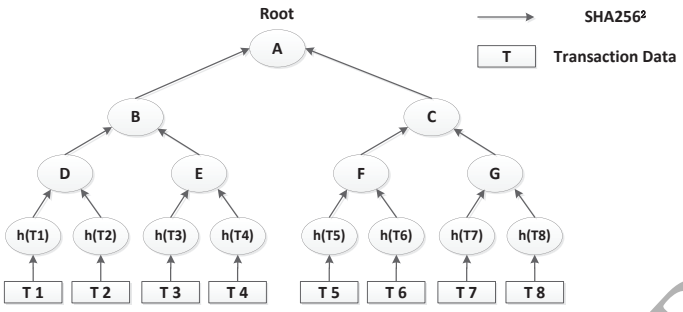

Figure 3: Structure of Merkle Hash Tree

be verified through its corresponding path. We can know whether the file data blocks' information in the MHT's leaf nodes are tampered or not by comparing their Merkle Root.

3.2. Files are chunked, encrypted and uploaded to P2P network

Normally, users' files are divided into 32MB per block in cloud storage and if the final block's size is less than 32MB, it will be filled with 0 [30]. Therefore, in our architecture, files are divided into 32MB per block before they are encrypted by the private key *sk* and uploaded to the P2P network. Moreover, we utilize a secure communication mechanism based on ECDSA, a kind of DSA as same as in Bitcoin. ECDSA [7] is the Digital Signature Algorithm implementing in Bitcoin and it is the combination of DSA and ECC proposed by Neal Koblitz [9] and Victor Miller [27]. The file blocks will be signed by the sender using the *ssk* and verified by the receiver using the *spk*.

In this architecture, the traditional PKC should be adopted but attribute-based encryption is unnecessary. Blockchain network is a P2P network without any third party or key generation center. Every user adopts an asymmetric encryption algorithm that users will keep their private key themselves and release their public key in public. Furthermore, the public key and private key here are not related to users' attributes.

3.3. Use blockchain as a trading mechanism

In financial field, Bitcoin has proven that it is possible to finish a trusted, auditable computing by utilizing a decentralized network and a public ledger. Unlike Bitcoin, transactions in our architecture are not strictly financial, instead they are collections of file storage location URLs, file hash values and etc. We assume that the blockchain's memories are tamper-proof under the same adversarial network environments as in Bitcoin. Because blockchain is an append-only distributed database, transactions are unchangeable once they are written in a block. So when retrospecting a transaction, we can just retrospect to the latest transaction as the file block latest status.

In this architecture, all users maintain a blockchain that stores all their files' and transactions' information, instead of the files themselves, when the transactions are settled in the blockchain. Unlike the identity-based scheme, the public key cannot be associated with the users' identity in the proposed architecture. When users verify their data, they can retrospect their transactions' information from the blockchain according to their identity and then verify their data through the file locations recorded by the transactions.

3.4. File storage strategy and file replicas replacement

The number of data replicas is not the essential factor here, and we assume that the number of replicas of each file block is known (e.g. 3). In traditional distributed cloud storage architectures, a file is divided into blocks of the same size and uploaded to the nearest distributed data center which is not thoroughly distributed. For example, the storage strategy in Hadoop is to store files and their replicas into one data center or storage server in triplicate in which one copy of the same virtual machine, one copy of the same host and one copy of the same rack. For security, we randomly place file blocks and their replicas around the users' nodes because malicious adversaries almost cannot get all blocks of a file which we will prove in Section 4.

3.5. File integrity verification

In Secure Hash Algorithm (SHA), the performance of hash operations depends on the length of the hashed message. SHA-256 used in Bitcoin is a kind of SHA that generates a 256 bits output from any input. The computation cost of file integrity verification will be very low because only the hash functions will be computed. SHA-256 is a hash operation with pseudo-randomness and its calculation process is irregular so that as long as the input has a slight change, its output will vary widely. This feature makes the MHT can detect any change about the transactions in all blocks though make every transaction as a leaf node in the MHT. In every block's head of the blockchain, we store the Merkle root computed by all transactions in the block, so that when we utilize the MHT to verify files' integrity we can just check whether the Merkle root is changed. Thus, all the transactions can be verified by the blockchain.

4. Network Performance & Security Analysis

In this section, we provide security analysis by comparing three kinds of cloud storage architectures (subsection A, B, and C) during the random process of users store and retrieve their files in the cloud. Our target aims to reduce network latency and maximum file loss rate of security events in the architectures. As for network performance, we consider transmission time as a criterion. Meanwhile, since the research works of consistency are almost based on the assumption that channels are reliable, and the blockchain consistency protocol is involved in this paper, thus we assume that all the channels are reliable.

We let L be the storage of file loss in the security issues, represented as follows.

$$L = LFS \cdot FLP. \tag{1}$$

The detail notations are listed in Table 1. In order to facilitate the comparison between architectures, we assume the fraction of malicious nodes is bounded by f $(0 \leq f < \frac{1}{3})$.

Table 1: Basic Notations for System Model

Notations	Meaning
LFS	The size of lost file
FLP	The probability of file loss
f	The fraction of malicious nodes
SAT	The successful attack times
m	The number of file blocks
n	The number of P2P network nodes
p	The probability of a successful attack among data centers
A	The users' files total sizes
$TFTT$	The total files transmission time
FTT	The file transmission time
bs	The size of a file block
bw_i	The bandwidth between the ith user and the data center
b_{ij}	The jth replica of the ith file block
$\|DC\|$	The number of data centers
$\|FBS\|$	The number of all file block replicas
bw_{ij}	The bandwidth between data center i and j
N_i	The ith node
U	The number of users
BWS_i	Links' latency of node N_i
FS_i	The size of transmitting file in the node N_i
r	The number of replicas of a file
K	The probability an honest node produces the next block
Q	The probability the attacker produces the next block
Q_z	The probability the attacker will ever catch up from z blocks behind

4.1. Multiple Users with Single Data Center

In this simple architecture, there is only one data center in the network where all users' data are stored in. Once the data center is broken down, all the users' file will be stolen and the users will suffer great losses. To calculate the latency, we simply ignore the links between the users and the data center. Because of the size of each file block is the same (32MB), the latency of transmitting a file block to the data center is $\frac{bs}{bw_i}$ (bw_i is variable) and the total network latency of this architecture is as follows.

$$\sum_{i=1}^{m} \frac{bs}{bw_i}. \tag{2}$$

The successful attack times SAT_1 ($0 \leq SAT_1 \leq |DC|$, all the data centers are broken down when $SAT_1 = |DC|$) of this architecture can be presented as follows.

$$SAT_1 = f \cdot U \cdot p. \tag{3}$$

According to (1) and (3), the loss storage of this architecture can be formulated as follows.

$$L_1 = A \cdot SAT_1. \tag{4}$$

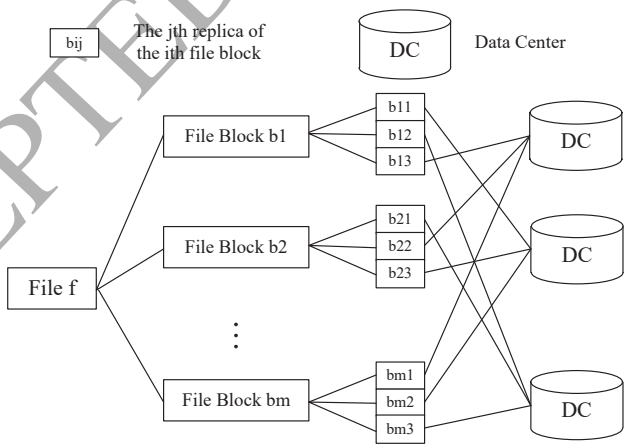

Figure 4: Architecture of Multiple Users with Multiple Data Centers & Genetic Algorithm

4.2. Multiple Users with Multiple Data Centers & Genetic Algorithm

For reducing the total files transmission time ($TFTT$) over users and data centers, we use a genetic algorithm to optimize this architecture [2]. As can be seen from Fig. 4, our genetic algorithm for multiple users and multiple data centers puts file block replicas respectively into different data centers, so that users can get their files quickly from the nearest data center. It can not only reduce the size of moving data, but also reduce the time of data movement and the number of movements.

4.2.1. File Block Replica Scheduling Coding Method

First, in our architecture, we mark all replicas of file blocks as b_{ij} which means the jth replica of file block b_i whose code is $\sum_{k=1}^{i-1} count_k + j$, and $count_k$ is the replica number of b_k. Users who download their files from cloud require a file block replica scheduling strategy code to illustrate the usage state of replicas of file blocks. Assume that a file block has n replicas, and we can utilize a n bit long 0/1 code to represent the requirement of every user downloading their files. If a user u_k needs a replica b_{ij}, the corresponding bit of b_{ij} is 1, otherwise is 0.

To better describe the strategy, we model the corresponding segment as the file block b_i coverage $Coverage(b_i)$. In our architecture, we suppose that if a user needs a file, he or she only needs to get one replica of every block of the file. For example, we assume there are two users u_1 and u_2 respectively with two files f_1 and f_2, and each file has three replicas, then the file block replica requiring code of f_1 can be 001000, 010000 or 100000, meanwhile the file block replica requiring code f_2 can be 001001, 010010 or 010100 and etc.

4.2.2. File Block Replica Distribution Coding Model

For data centers, we use a $|log_2|DC||$ long binary bits to represent them where $|DC|$ is the number of data centers. For example, where $|DC| = 10$, there are codes 0001, 0010, 0011, 1000, 1001, 0000, they represents $dc_1, dc_2, dc_3, \ldots, dc_9$, dc_{10}, respectively. All file block replicas of users are defined as FBS in

our architecture, and we use a $|FBS| \cdot |log_2|DC||$ binary bits to represent a file block replica distribution solution. In a file block replica distribution solution, we let the chromosome code be $Ccode$ for $i = 1, 2, \ldots, |FBS|$, if $\sum_{m=1}^{k-1} count_m < i < \sum_{m=1}^{k} count_m$, the ith gene segment in the left side of the chromosome illustrates the number of data centers which the $i - \sum_{m=1}^{k-1} count_k$th replica of file block b_k is stored. Actually, not every binary number generated above can indicate a valid distribution solution. If there existing at least one segment in chromosome can't represent a data center, it's an invalid solution.

4.2.3. Validation Method

Our file block placement strategy is based on a genetic algorithm and its procedure including mutation stage and crossover stage will generate valid solutions and invalid solutions. Thus, we propose a validation method to verify whether the solution is valid or not and eliminate invalid solutions. For user u_i who needs file block b_i, there is one and only one mapping from u_i to b_i. Meanwhile, it's invalid if there are more than one file block replica stored in the same data center because it will waste the storage space and increase the network redundancy.

4.2.4. Evaluation Function for File Block Replica Placement Strategy

The purpose of this placement strategy is to decrease the $TFTT$ over users and data centers. Therefore, we take the global FTT as an evaluation function. The global FTT is made up of all transmission time during the process of cloud storage in this architecture. All file replicas are scheduled to the nearest data center and then transmitted to users. The nearest here denotes that the transmission time is the smallest.

Every file stored in the cloud must meet at least two conditions: (a) the file block replicas are scheduled to the nearest data center, (b) file block replicas required by a user are in the same data center. Therefore, if file block replicas of a file are in different data centers, it's necessary to schedule them to the same data center before they are downloaded by users. The file block scheduling is

not the essential factor in this paper and has been researched in many previous works, thus we suppose that the cost of file block scheduling is negligible and far less than the cost of data transmission.

Considering the data transmission time of a single transmission of a file block replica between data center dc_i and dc_j, it can be represented as follows.

$$SingleTimeCost(b_k, dc_i, dc_j) = \frac{bs_k}{bw_{ij}} + C_{ij}, \tag{5}$$

where bs_k indicates the size of file block b_k, bw_{ij} is the bandwidth between data center dc_i and dc_j. Furthermore, procedures like connecting a link, request, respond and disconnecting a link during transmissions will result the share of time cost we denote as C_{ij}. Actually, data size of a file stored in cloud is usually huge but C_{ij} is very small, so formula (5) can be simplified as follows.

$$SingleTimeCost(b_k, dc_i, dc_j) \approx \frac{bs_k}{bw_{ij}}. \tag{6}$$

Finally, according to (6), the global FTT can be calculated approximately in the follow formula.

$$\sum_{k=1}^{|DC|} SingleTimeCost(b_k, dc_i, dc_j) \approx \sum_{k=1}^{|DC|} \frac{bs_k}{bw_{ij}}. \tag{7}$$

4.2.5. Security Analysis

Considering physical natural disasters and hacker attacks, a security parameter η [48] is given in this architecture for the measurement and we assume this architecture is loading balance that users' files are evenly distributed among the data centers. The file loss in this architecture can be measured by the number of data centers $|DC|$, the data size of all files A, the probability of a successful attack among data centers p and a given security parameter η. Assuming that, the random process in this architecture is a Poisson distribution: $\lambda_1 = \frac{A}{|DC|} \cdot p$ (see Table 1). The successful attack times SAT_2 ($0 \leq SAT_2 \leq |DC|$, all the data centers are broken down when $SAT_2 = |DC|$) of this architecture can be presented as follows.

$$SAT_2 = \sum_{k=1}^{f \cdot U} \frac{{\lambda_1}^k \cdot e^{-\lambda_1}}{k!}. \tag{8}$$

Therefore, according to (1) and (2), the loss function in this architecture can be calculated in the follow formula.

$$L_2 = SAT_2 \cdot \frac{A}{|DC|} \cdot \eta. \tag{9}$$

4.3. Multiple Users with Multiple Data Centers & Blockchain

4.3.1. Network Performance Analysis

In our architecture, users can either be a client or a server and users who have vacant disk storage space can provide their space to users who need space. For improving network performance, users can randomly select a node that the transmission latency is the smallest. All nodes are encoded as $N_1, N_2, \ldots, N_n$ in this architecture, and links' latency of the ith node N_i can be represented as BWS_i, the size of transmitting files in the ith node N_i can be represented as FS_i. Therefore, the FTT of the ith node is $\frac{FS}{\arg minBWS_k}$, and the $TFTT$ is the sum of FTT as follows.

$$\sum_{k=1}^{n} \frac{FS}{\arg\min_k BWS_k}. \tag{10}$$

4.3.2. Security Analysis

The security of blockchain has been verified through Bitcoin and proofed in many other previous works. Unless more than half of computational power in the network is under malicious adversaries' control, contents of transactions are unchangeable because those malicious adversaries can build a longer branch and users only accept the longer one. Meanwhile, repeat transactions can be avoided since transactions are broadcasting to the entire network as soon as they are done.

In our architecture, file block replicas are randomly stored in the nodes nearby, and blockchain is utilized as a trading mechanism among users. The probability of malicious adversaries getting a file block replica is $\frac{r}{n}(r = 1, 2, 3, \ldots)$, but the probability of getting all m blocks of a file is $(\frac{r}{n})^m$. If n and m are big enough, the probability of malicious adversaries getting all file blocks or file replicas is infinitely tending to 0.

Assuming that the attacker possesses more than half of computational power in the network, they can forge a longer chain that consisting fake trading information. Because all users only recognize the longest chain, an attacker can cheat or make chaos depending on the proportion of computational power they possess. In paper [29], Nakamoto had proofed the following formula.

$$Q_z = \begin{cases} 1, & \text{if } K \leq Q \\ (\frac{Q}{K})^z, & \text{if } K > Q \end{cases} \tag{11}$$

Assuming that K is larger than Q, the successful probability of an attack is an exponential decline due to the increase in the number of blocks. The attackers' potential progress will be a Poisson distribution: $\lambda_2 = z \cdot \frac{Q}{K}$ (see Table 1) under the condition of honest nodes spend average expected time generating per block. Therefore, according to (11), the probability of attackers catching up (e.t. the successful attack times) can be calculated in the following formula, where $g(k) = \frac{{\lambda_2}^k \cdot e^{-\lambda_2}}{k!}$.

$$SAT_3 = \begin{cases} \sum_{x=0}^{k} g(x) \cdot (\frac{Q}{K})^{z-k}, \text{if } k \leq z \\ \sum_{x=0}^{\infty} g(x) \qquad\qquad , \text{if } k > z \end{cases} \tag{12}$$

To avoid summing the infinite tail of the distribution as same as in Bitcoin, the formula (12) can be simplified as follows.

$$SAT_3 = 1 - \sum_{k=0}^{z} g(k) \cdot (1 - (\frac{Q}{K})^{z-k}). \tag{13}$$

Finally, according to (1) and (13), the file loss function of our architecture is as follows.

$$L_3 = (\frac{r}{n})^m \cdot SAT_3 \cdot \frac{32}{1024}. \tag{14}$$

5. Experiment & Analysis

In this section, we evaluate the performance of our architecture with extensive numerical analysis. Evaluation methods have been proposed in Section 4,

and the evaluations simulate the processes of users store and retrieve their files in the cloud. In subsection 5.1 we introduce the environment setup and parameter setting, and in subsection 5.2 we analyze the results of these evaluations.

5.1. Simulation Setup

The experiments are implemented in Java 1.8 on a desktop PC running Windows 7 with Inter(R) Pentium(R) G3460 3.50GHz, RAM 4GB, DISK 1TB, Bandwidth 100MB/s. First, we evaluate the influence of the number of users on performance of network, then the influence of the number of file block replicas on performance of network, and finally the influence of the number of users on the loss of files.

In our experiment, unless otherwise stated, we assume the storage capacity of each data center is 1000GB and the storage capacity of each node is randomly varying from 17GB to 26GB. The bandwidth between data centers and nodes, data centers and data centers are randomly between 5MB/s to 10MB/s. The results depicted in the following figures are averaged for 10 repeated executions.

5.2. Evaluation Results

In our evaluation, we will evaluate the network performance and security by using the formulas given in Section 4.

5.2.1. The Number of Users and Network Latency

Network latency is utilized to be the criterion of the network performance in these architectures, and it's mainly decided by the size of transmitted files. Generally speaking, the size of transmitted files is strongly relative to the number of users and the number of file block replicas. Transmission time in this evaluation denotes that the time spending on storing and retrieving files on the cloud.

When the number of users is small, the architecture of Multiple Users with Multiple Data Centers & Genetic Algorithm will spend more time on calculating the genetic algorithm, while the others do not. Meanwhile, in the architecture

of Multiple Users with Single Data Center, users need to calculate which link to the data center is the fastest. But in the architecture of Multiple Users with Multiple Data Centers & Blockchain, users directly store and retrieve their files in nodes around them which has the smallest transmission time. As shown in Fig. 5, when the number of users is small, network latency in the architecture proposed in this paper is the lowest.

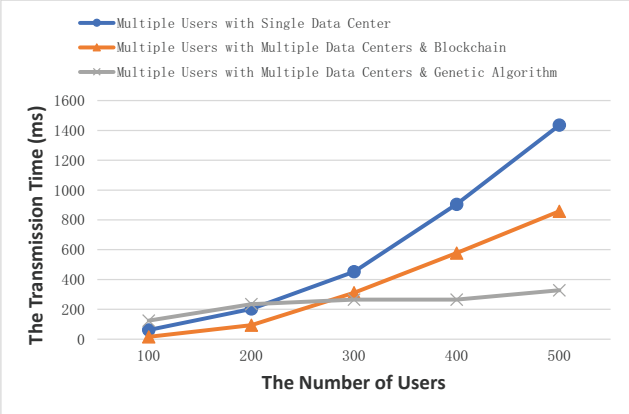

Figure 5: The Transmission Time with Different Number of Users

With the increasing numbers of users, the amount of calculation in the architecture of Multiple Users with Multiple Data Centers & Genetic Algorithm is not too large compared to others and the solutions output from the genetic algorithm are approximately optimal. In these solutions, users' file block replicas are stored in the nearest data center and the transmission time becomes smallest among the architectures. In the architecture of Multiple Users with Multiple Data Centers & Blockchain, the cost of computing in blockchain getting larger, but it still takes advantage of the files distributing around users. And in the architecture of Multiple Users and Single Data Center, the computational time in selecting the fastest link becomes larger, and it has the largest transmission time. As can be seen in Fig. 5, the transmission delay on the proposed architecture is averagely reduced by 39.28% and 76.47% on the number of users and on the number of file block replicas, respectively, in comparison to the architecture

with single data center.

5.2.2. The Number of File Replicas and Network Latency

To evaluate the performance of network in different number of data replicas, the number of data centers in the architecture of Multiple Users with Multiple Data Centers & Genetic Algorithm is set to 10 and there are 1200 files to store. Besides, the results depicted in Fig. 6 are averaged values evaluating under the number of users ranging from 100 to 500. In each architecture, the network latency will be calculated by formulas (2), (7), (10) in Section 4.

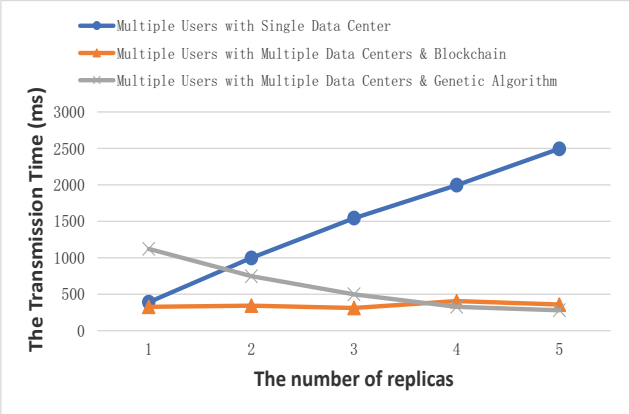

Figure 6: The Transmission Time with The Number of Replicas

In the architecture of Multiple Users with Single Data Center, it has only one data center and more replicas will lead to more serious network jamming. Thus, the number of file block replicas is meaningless in this architecture.

In the architecture of Multiple Users with Multiple Data Centers & Genetic Algorithm, if the number of file block replicas is too small, the scheduling of replicas among the data centers and nodes will spend a lot of time. Meanwhile, if the number of file block replicas is too large, network redundancy will be high although the transmission time is reducing. As we can see in Fig. 6, the transmission time is reduced more than 50% when the number of replicas grows from 1 to 3, but the reduction of transmission time becomes smaller when it

grows larger. In conclusion, the number of file block replicas should be set to 3 in this architecture.

In our architecture, the number of replicas almost has no influence on the transmission time due to the replicas are stored in nodes around. As can be seen in the Fig. 6, the transmission delay on the number of file block replicas is averagely reduced by 41.36% than that on the distributed architecture using the genetic algorithm.

5.2.3. The Number of Users and File Security

In this experiment, the rate of the stolen file is utilized as the criterion of security. As the storage methods of each architecture are different, the storage of lost file is different either. Thus, the total storage of each architecture is equally set to 10000GB.

The given security parameter η and the probability of security issues among data centers p are inputs of the architectures, and they are numbers very closed to 1. In our experiment, the security parameter η is set to 0.9, the probability of a security issue is set to 0.1, $|DC|$ is set to 10, r is set to 3 and z is set to 5. Considering the extreme situation, the number of malicious nodes is set to $\lfloor U \cdot \frac{1}{3} \rfloor$ due to $0 \leq f < \frac{1}{3}$. K is set to 0.9 and Q is set to 0.1. In each architecture, the storage of file loss will be calculated by formulas (4), (9), (14) proposed in Section 4.

As we can see in the Fig. 7, on average, the file loss rate in this work is almost 0% based on the assumptions made in the simulation for our architecture while it's nearly 100% and 71.66% for the architecture with single data center and the distributed architecture using the genetic algorithm. In the architecture of Multiple Users with Single Data Center, all files will be stolen once the only data center is broken down. In the architecture of Multiple Users and Multiple Data Centers & Blockchain, the storage of file loss is almost 0 because that the probability of successful attack is very close to 0 and every successful attack only steal or tamper a 32MB file block data. As the architecture of Multiple Users with Multiple Data Centers & Genetic Algorithm, with the number of

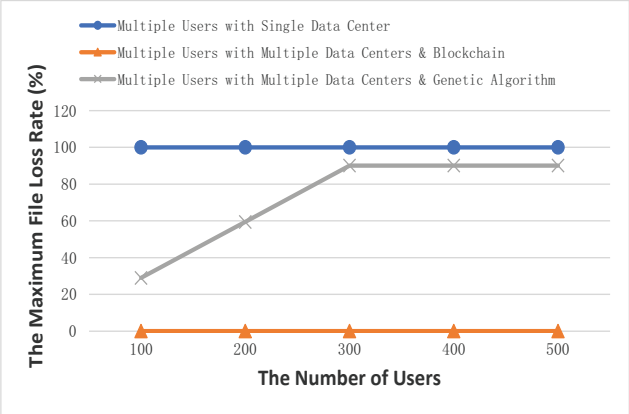

Figure 7: The Maximum File Loss Rate with The Number of Users

users increasing, the storage of file loss is getting larger and when the number of users is bigger than 300, the maximum file loss rate is unchangeable due to the limited condition $0 \leq SAT_2 \leq |DC|$.

In conclusion, no matter how many attacks, the architecture proposed in this paper has an outstanding performance in security. The security performance is relative low on the architecture of Multiple Users with Single Data Center. With the number of users increases, the security performance in the architecture of Multiple Users with Multiple Data Centers & Genetic Algorithm is becoming relatively low.

6. Conclusion

This paper has proposed a blockchain-based security architecture for distributed cloud storage. The proposed architecture has been compared with other two traditional architectures in terms of security and network transmission delay. Based on the simulation assumptions utilized in this paper, the file loss rate of the proposed architecture outperforms other two traditional architectures on average. Meanwhile, the network performance of the traditional distributed architecture has been improved by the customized genetic algorithm which reduces the costs on replicas scheduling and transmitting. And the trans-

mission delay on the proposed architecture is lower than other two traditional architectures. Comparative simulation results illustrate that the proposed architecture has outstanding security performance and network performance.

Acknowledgement

Part of the work was presented in IEEE International Symposium on Parallel and Distributed Processing with Application (ISPA) (2017). This work was supported by the National Key R&D Program of China (2018YFB1003201), the National Natural Science Foundation of China (61672171, 61702115, 61702114), Major Research Project of Educational Commission of Guangdong Province (2016KZDXM052), R&D Project of Guangdong Province (2017B030305003), Opening Project of Guangdong Province Key Laboratory of Big Data Analysis and Processing (201805).

References

[1] Wenbin Chen, Hao Lei, and Ke Qi. Lattice-based linearly homomorphic signatures in the standard model. *Theoretical Computer Science*, 634:47–54, 2016.

[2] Lizhen Cui, Junhua Zhang, Lingxi Yue, Yuliang Shi, Hui Li, and Dong Yuan. A genetic algorithm based data replica placement strategy for scientific applications in clouds. *IEEE Transactions on Services Computing*, pages 1–13, 2017. doi:10.1109/TSC.2015.2481421.

[3] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[4] Hoang Giang Do and Wee Keong Ng. Blockchain-based system for secure data storage with private keyword search. In *Services (SERVICES), 2017 IEEE World Congress on*, pages 90–93. IEEE, 2017.

[5] Qiang Fu, Jörg-Uwe Pott, Feng Shen, and Changhui Rao. Stochastic parallel gradient descent optimization based on decoupling of the software and hardware. *Optics Communications*, 310:138–149, 2014.

[6] Shay Gueron, Simon Johnson, and Jesse Walker. Sha-512/256. In *Eighth International Conference on Information Technology: New Generations*, pages 354–358, 2011.

[7] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security*, 1(1):36–63, 2001.

[8] Mariam Kiran and M Stanett. Bitcoin risk analysis. http://www.nemode.ac.uk/wp-content/uploads/2015/02/2015-Bit-Coin-risk-analysis.pdf, 2015.

[9] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[10] Bo Li, Yanyu Huang, Zheli Liu, Jin Li, Zhihong Tian, and Siu-Ming Yiu. Hybridoram: practical oblivious cloud storage with constant bandwidth. *Information Sciences*, 2018. doi:10.1016@j.ins.2018.02.019.

[11] Hongwei Li, Rongxing Lu, Liang Zhou, Bo Yang, and Xuemin Shen. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, 8(2):655–663, 2014.

[12] Jiaxing Li, Zhusong Liu, Long Chen, Pinghua Chen, and Jigang Wu. Blockchain-based security architecture for distributed cloud storage. In *Ubiquitous Computing and Communications (ISPA/IUCC), 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on*, pages 408–411. IEEE, 2017.

[13] Jin Li, Xiaofeng Chen, Sherman SM Chow, Qiong Huang, Duncan S Wong, and Zheli Liu. Multi-authority fine-grained access control with account-

ability and its application in cloud. *Journal of Network and Computer Applications*, 112:89–96, 2018.

[14] Jin Li, Xiaofeng Chen, Xinyi Huang, Shaohua Tang, Yang Xiang, Mohammad Mehedi Hassan, and Abdulhameed Alelaiwi. Secure distributed deduplication systems with improved reliability. *IEEE Transactions on Computers*, 64(12):3569–3579, 2015.

[15] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick PC Lee, and Wenjing Lou. A hybrid cloud approach for secure authorized deduplication. *IEEE Transactions on Parallel and Distributed Systems*, 26(5):1206–1216, 2015.

[16] Jingwei Li, Jin Li, Dongqing Xie, and Zhang Cai. Secure auditing and deduplicating data in cloud. *IEEE Transactions on Computers*, 65(8):2386–2396, 2016.

[17] Tong Li, Jin Li, Zheli Liu, Ping Li, and Chunfu Jia. Differentially private naive bayes learning over multiple data sources. *Information Sciences*, 444:89–104, 2018.

[18] Qun Lin, Jin Li, Zhengan Huang, Wenbin Chen, and Jian Shen. A short linearly homomorphic proxy signature scheme. *IEEE Access*, 6:12966–12972, 2018.

[19] Qun Lin, Hongyang Yan, Zhengan Huang, Wenbin Chen, Jian Shen, and Yi Tang. An id-based linearly homomorphic signature scheme and its application in blockchain. *IEEE Access*, 2018. doi:10.1109@ACCESS.2018.2809426.

[20] Bin Liu, Xiao Liang Yu, Shiping Chen, Xiwei Xu, and Liming Zhu. Blockchain based data integrity service framework for iot data. In *Web Services (ICWS), 2017 IEEE International Conference on*, pages 468–475. IEEE, 2017.

[21] Zheli Liu, Chuan Fu, Jun Yang, Zhusong Liu, and Lingling Xu. Coarser-grained multi-user searchable encryption in hybrid cloud. In *Transactions*

on Computational Collective Intelligence XIX, pages 140–156. Springer, 2015.

[22] Zheli Liu, Yanyu Huang, Jin Li, Xiaochun Cheng, and Chao Shen. Divoram: Towards a practical oblivious ram with variable block size. *Information Sciences*, 447:1–11, 2018.

[23] Zheli Liu, Tong Li, Ping Li, Chunfu Jia, and Jin Li. Verifiable searchable encryption with aggregate keys for data sharing system. *Future Generation Computer Systems*, 78:778–788, 2018.

[24] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30. ACM, 2016.

[25] Rishi Lyengar. Apple to strengthen security after icloud nude celebrity photos leak. http://time.com/3271667/apple-jennifer-lawrence-icloud-leak-security/, 2014. Accessed September 4, 2014.

[26] Weizhi Meng, Elmar Tischhauser, Qingju Wang, Yu Wang, and Jinguang Han. When intrusion detection meets blockchain technology: a review. *IEEE Access*, 2018. doi:10.1109@ACCESS.2018.2799854.

[27] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.

[28] Rekha Nachiappan, Bahman Javadi, Rodrigo N Calheiros, and Kenan M Matawie. Cloud storage reliability for big data applications: A state of the art survey. *Journal of Network and Computer Applications*, 97:35–47, 2017.

[29] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`, 2008.

[30] T. C. Nguyen, W. Shen, Z. Lei, W. Xu, W. Yuan, and C. Song. A probabilistic integrity checking approach for dynamic data in untrusted cloud storage. In *2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*, pages 179–183, June 2013. doi:10.1109/ICIS.2013.6607837.

[31] Manoj Parameswaran, Anjana Susarla, and Andrew B Whinston. P2p networking: an information sharing alternative. *Computer*, 34(7):31–38, 2001.

[32] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques*, pages 47–53. Springer, 1984.

[33] Han Shen, Chongzhi Gao, Debiao He, and Libing Wu. New biometrics-based authentication scheme for multi-server environment in critical systems. *Journal of Ambient Intelligence and Humanized Computing*, 6(6):825–834, 2015.

[34] Heng Tao Shen, Yanfeng Shu, and Bei Yu. Efficient semantic-based content search in p2p network. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):813–826, 2004.

[35] Jian Shen, Ziyuan Gui, Sai Ji, Jun Shen, Haowen Tan, and Yi Tang. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *Journal of Network and Computer Applications*, 106:117–123, 2018.

[36] Jian Shen, Chen Wang, Tong Li, Xiaofeng Chen, Xinyi Huang, and Zhi-Hui Zhan. Secure data uploading scheme for a smart home system. *Information Sciences*, 2018. doi:10.1016/j.ins.2018.04.048.

[37] Jian Shen, Tianqi Zhou, Xiaofeng Chen, Jin Li, and Willy Susilo. Anonymous and traceable group data sharing in cloud computing. *IEEE Transactions on Information Forensics and Security*, 13(4):912–925, 2018.

[38] Eno Thereska, Hitesh Ballani, Greg O'Shea, Thomas Karagiannis, Antony Rowstron, Tom Talpey, Richard Black, and Timothy Zhu. Ioflow: a software-defined storage architecture. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pages 182–196. ACM, 2013.

[39] Hao Wang, Zhihua Zheng, Lei Wu, and Ping Li. New directly revocable attribute-based encryption scheme and its application in cloud storage environment. *Cluster Computing*, 20(3):2385–2392, 2017.

[40] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li. Enabling public auditability and data dynamics for storage security in cloud computing. *IEEE transactions on parallel and distributed systems*, 22(5):847–859, 2011.

[41] Shawn Wilkinson, Tome Boshevski, Josh Brandoff, and Vitalik Buterin. Storj a peer-to-peer cloud storage network. 2014.

[42] Changsong Yang, Xiaofeng Chen, and Yang Xiang. Blcokchain-based publicly verifiable data deletion scheme for cloud storage. *Journal of Network and Computer Applications*, 103:185–193, 2017. doi:10.1016/j.jnca.2017.11.011.

[43] Yong Yu, Man Ho Au, Giuseppe Ateniese, Xinyi Huang, Willy Susilo, Yuanshun Dai, and Geyong Min. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. *IEEE Transactions on Information Forensics and Security*, 12(4):767–778, 2017.

[44] Rui Zhang, Chuang Lin, Kun Meng, and Lin Zhu. A modeling reliability analysis technique for cloud storage system. In *Communication Technology (ICCT), 2013 15th IEEE International Conference on*, pages 32–36. IEEE, 2013.

[45] Xiaosong Zhang, Yuan Tan, Chen Liang, Yuanzhang Li, and Jin Li. A covert channel over volte via adjusting silence periods. *IEEE Access*, 2018. doi:10.1109@ACCESS.2018.2802783.

[46] Yinghui Zhang, Xiaofeng Chen, Jin Li, Duncan S Wong, Hui Li, and Ilsun You. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Information Sciences*, 379:42–61, 2017.

[47] Yinghui Zhang, Dong Zheng, and Robert H Deng. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet of Things Journal*, 2018. doi:10.1109/JIOT.2018.2825289.

[48] Lan Zhou, Vijay Varadharajan, and Michael Hitchens. Achieving secure role-based access control on encrypted data in cloud storage. *IEEE Transactions on Information Forensics & Security*, 8(12):1947–1960, 2013.

[49] Xiaoming Zhu, Bingying Song, Yingzi Ni, Yifan Ren, and Rui Li. Software defined anythingfrom software-defined hardware to software defined anything. In *Business Trends in the Digital Era*, pages 83–103. Springer, 2016.

[50] Guy Zyskind, Oz Nathan, et al. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*, pages 180–184. IEEE, 2015.