Using Blockchain for Data Auditing in Cloud Storage: 4th International Conference, ICCCS 2018, Haikou, China, June 8–10, 2018, Revised Selected Papers, Part III

Chapter · September 2018

DOI: 10.1007/978-3-030-00012-7_31

CITATIONS
2

READS
466

5 authors, including:

Chunhua Li
Huazhong University of Science and Technology
14 PUBLICATIONS **77** CITATIONS
SEE PROFILE

Ke Zhou
Huazhong University of Science and Technology
146 PUBLICATIONS **980** CITATIONS
SEE PROFILE

Using Blockchain for Data Auditing in Cloud Storage

Chunhua Li(✉), Jiaqi Hu, Ke Zhou, Yuanzhang Wang, and Hongyu Deng

Wuhan National Lab for Optoelectronics,
Huazhong University of Science and Technology, Wuhan 430074, China
li.chunhua@hust.edu.cn

Abstract. Cloud storage is one of the most important service of cloud computing. Since cloud service providers can not be completely trusted, traditional auditing methods can't guarantee the security of data sources. This paper proposes a security framework for cloud data audit using blockchain technology. User's operational information on the file is formed to a block after validated by all checked nodes in the blockchain network, and then to be put into the blockchain. Any modification or fake to the operational information can be inspected through the chain structure of block, thus ensuring the security of auditing data source. We construct a prototype in an Ethereum-based blockchain using Aliyun as data storage service, then test the time overhead of uploading file, broadcasting operation information and packing information into block chain. The results show that the time for packaging block remains unchanged from an overall viewpoint, and as the file size increasing, packaging block occupies less percentage in the entire process of file uploading or downloading.

Keywords: Cloud storage · Behavior auditing · Customizable log Blockchain

1 Introduction

In the cloud environment, data owner and user often rely on a trusted third party for authentication and authorization. However, a third party is not secure by nature. Some security incidents have occurred repeatedly in recent years, such as data leakage and data tampering. A third party may reveal user's data for economy benefit actuation, on the other hand, some users may maliciously declare data loss for high compensation. Due to the lack of mutual trust, more than 70% of companies are not planning to adopt cloud storage services in the near future. Therefore, an audit scheme based on a trusted architecture becomes increasing important in the cloud [2, 3].

Log analysis is a common method in many auditing schemes [4], which track data through extracting users' operation events from system log. However, it is inefficiency to analyze users' operation from large amounts of system log records. Itani et al. used hash chains to maintain the order consistency of operation records [5]. All these methods count on system log to provide them reliable data operation records. In fact, it is hard to guarantee cloud server provider (CSP) believable. Ateniese et al. proposed a

X. Sun et al. (Eds.): ICCCS 2018, LNCS 11065, pp. 335–345, 2018.
https://doi.org/10.1007/978-3-030-00012-7_31

called PDP (provable data possession) approach, which can verify the integrity of outsourced data in untrusted cloud environments by introducing third-party auditors (TPA) and supports sample auditing [6]. Tian et al. proposed a public auditing for users' operation behaviors in cloud storage [7], in which a trusted third party is introduced to verify the integrity of operation behavior logs to enhance the credibility of forensic results. But, TPA is not as dependable as you might expect, it may collude with CSP or users, or be prone to attacks such as tampering and forgery.

The emergence of blockchain technology provides a new research idea to solve the problem of mutual trust. It utilizes cryptography rather than centralized architecture to build trust in peers for safeguarding interactions of them [8]. Meanwhile, it employs consensus algorithm to generate and update data between peers to ensure that block data is not changed, thus very suitable for data security in Cloud. In the past two years, some cloud security schemes based on blockchain have been proposed. Sengupta et al. proposed a scheme called Retricoin which replaces the heavy computational proof-of-work of Bitcoin by proofs of retrievability [9]. To guarantee the availability of an important but large file, they distributed the file segments among the users in the Bitcoin network. Ramachandran et al. used blockchain to develop a secure and immutable scientific data provenance management framework [10], in which utilizes smart contacts to record immutable data, and efficiently prevent any malicious modification to the captured data. Yang et al. proposed a public verifiable data deletion scheme for cloud storage based on blockchain, which uses the idea of blockchain to guarantee that any malicious deletion operation can be verified [11]. Dagher et al. proposed a blockchain-based framework for secure, interoperable and efficient access to medical records, which utilizes smart contracts in an Ethereum-based blockchain for heightened access control, and employs advanced cryptographic techniques for further security [12]. Ghoshal et al. proposed an auditing mechanism using the blockchain data structure of Bitcoins [13], any user can perform the validation of selected files efficiently. Fu et al. proposed a blockchain-based secure data-sharing protocol under decentralized storage architecture [14]. The mentioned schemes above can be regarded as application trial of blockchain in cloud security. Since the capacity of a block is limited in the blockchain, only very important security information is considered to store into the block, or system performance will not be acceptable.

In this paper, we try to use blockchain to ensure the security of audit data source. We construct a security analysis framework in an Ethereum-based blockchain, and use Aliyun as data storage service. The key contributions of this paper can be summarized as follows:

- We design a block structure for auditing users' operation. The file metadata and operation information is put into a block called log block, which is broadcasted among peers in the network and validated by all checked nodes with a lightweight consensus algorithm. All verified log blocks is chained through the hash of adjacent blocks. The chained metadata information can later be used for data integrity verification, the chained operation record can later be used for tracing access to file.
- We design a security architecture based on blockchain network which includes four components: blockchain layer, proxy layer, user layer and cloud storage layer. We implement a prototype in an Ethereum-based blockchain using Aliyun as data storage service. The results of analysis and tests demonstrate that our scheme is feasible and can resist repudiation attacks and replay attacks.

Paper organization is as follows: Sect. 2 introduces the related knowledge of blockchain technology, and analyzes its architecture and security. Section 3 describes our design in detail, and gives a simple security analysis. The performance evaluation is shown in Sect. 4. The conclusion is drawn in Sect. 5.

2 Backgrounds

2.1 Blcokchain's Characteristics and Security

Blockchain is a novel distributed technology of verifying and storing data using an encrypted chain block structure [1, 3]. It is a public distributed ledger that can be shared, replicated, and synchronized among different nodes. Combining cryptographic algorithms, decentralized consensus mechanism and P2P network, blockchain provides a way for all nodes in the network to reach the same state in a secure and verifiable manner.

***Decentration*:** The blockchain network adopts P2P network in which all nodes are in the equal status. All data distributed in blockchain network is available for any node, and newly added node can select to download all or part of the block data from the old nodes to query or verify the block data. Each transaction which generated in the network is broadcasted to all nodes and verified and updated by the miner nodes. With the support of blockchain technology, the storage of cloud data is no longer dependent on a small number of data centers, it can be distributed into much more nodes, thus preventing one or more data centers from being attacked or improperly managed. At the same time, bad behavior such as data loss or disclosure can be discovered in time.

***Tamper-Resistant*:** The block structure in the blockchain network is shown in Fig. 1. Each block contains the hash value of the previous block, thus forming a complete chain structure in the network. The pre-hash in block header is computed from the previous block using a specific hash function. Some necessary information is also stored in the block header for the purpose of security verification, such as timestamp, the difficulty of the puzzle and the Merkle root. Suppose a malicious user change the data of the previous block, it will inevitably cause the hash change of this block, furtherly result in the inconsistency between the hash of previous block and the pre-hash of current block. When two miners construct a new block at the same time, both blocks will be linked to the current block. Once the blockchain network is forked, the blockchain always trusts the longest chain. Generally speaking, unless malicious node has mastered more than 51% power of the entire network for a long time, it is impossible to replace the normal chain and complete data modification, otherwise resulting in extremely high tampering costs. Therefore, blockchain can rely on proof of work (POW) and consensus algorithms to safeguard the system.

2.2 Merkle Hash Tree

A Merkle Hash Tree (MHT) [15] is a well-studied authentication structure. It is often constructed as a binary tree where the leaf nodes store the hashes of data elements

(a file or a collection of files) and the non-leaf nodes store the hashes of its two children. Generally, MHT is used to identify whether the data has been altered or not by comparing the computed root hash and the one verifier holds. In blockchain network, MHT is also employed to store transaction's hash so as to check transaction's authenticity.

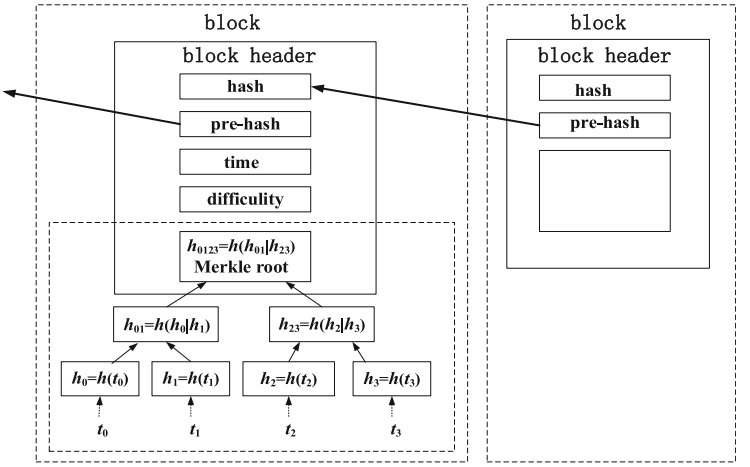

Fig. 1. Block structure in blockchain

As Fig. 1 shows, each block header saves the root hash of all transaction data in this block. When the verifier with a root h_{0123} wants to verify the transaction t_1, the prover will send him the auxiliary verification information $\{h_0, h_{23}\}$. Then the verifier computes $h_1 = h(t_1)$, $h_{01} = h(h_0| h_1)$, $h_{0123} = h(h_{01}| h_{23})$, and check if the computed root h_{0123} is the same as the one he holds. If there are an odd number of transactions, the remaining one is hashed with itself. Further, root hash participates in the hash operation of block header, thus any modification to transaction data will lead to the change of the root hash, which will result in the hash change of the block header. In this paper, we further utilize MHT to construct a hash tree for file metadata and file operation respectively, and make use of blockchain's decentralized architecture to ensure the security of audit data source.

3 Design and Implementation

The goals of our auditing framework is mainly to prevent log information from being modified by attackers or CSP. In addition, the process of adding a new block into the block chain will consume a lot of time, we'll try our best to reduce the latency.

3.1 System Architecture

We use blockchain to store user's operation on the file and metadata information when the file is uploaded. The system does not care about the actual location of a file, it only stores a file URL in file metadata. We utilize the tamper-resistant nature of blockchain to ensure the reliability of operation logs and file metadata. Metadata information can later be used to conduct integrity auditing, behavior auditing can be conducted by analyzing the operation logs.

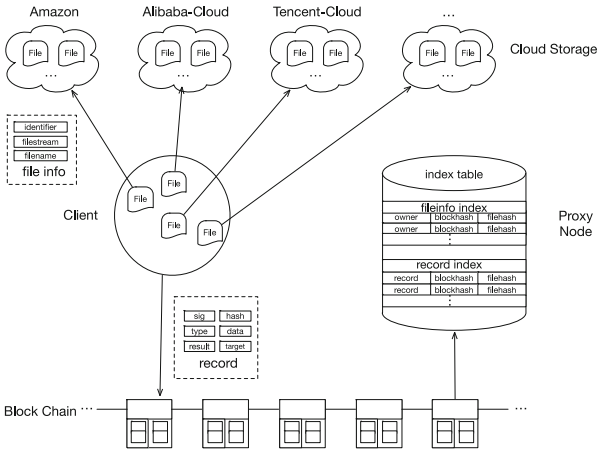

Fig. 2. System architecture

In order to accelerate the query of block data, we introduce a proxy layer. The proxy layer stores the index of block data. Users can quickly locate the requested block by interacting with the proxy layer. Of course, the proxy layer is not necessary, and a user can also traverse the block chain to get information. In order to accelerate block packaging, we plan to introduce lightning network into our system. Lightning network is a newly proposed technology which is scalable off-chain instant payments. However we are still finding a way to make it do with lightning network.

Figure 2 shows our system architecture, it is composed of four parts: blockchain layer, proxy layer, user layer and cloud storage layer.

Blockchain Layer: It consists of various nodes on the blockchain network, each node equally accepts the operation information broadcasted by the user node. The record information is packaged into the block by the mining algorithm. In the system, we use lightweight consensus protocol, that is, only a part of the miner is designated to participate in mining operation, while the mining interface is accessible to all nodes.

Proxy Layer: It is a node in the network used to keep the index of all blocks to speed up the query. When a newly generated block is put to the proxy node, it will analyze the content of block and forms a record index. When users need to get their own files or historical operation records, the proxy node can quickly locate the corresponding blocks according to the index.

User Layer: It is responsible for sending requests to the cloud and broadcasting operation information to each miner node of blockchain layer. When the operation record needs to be reviewed, the block index can also be obtained from the proxy node to retrieve related data.

Cloud Storage Layer: It also called persistent storage, namely, an actual storage location of data. In my system, only data is stored in cloud, file metadata is kept in the blockchain.

3.2 Data Structure for Auditing

Existing security frameworks based on blockchain are designed for their own security goal, so block information, consensus algorithm and data flow used in these system are different. Considering that the capacity of a block is limited, there may not have enough space to store the actual data in block. So, we only store the most important metadata which reflects the initial status of data or user behavior. From Fig. 3 we can see that such metadata information as file storage address, file hash, and the owner of a file will be kept according to our auditing goal.

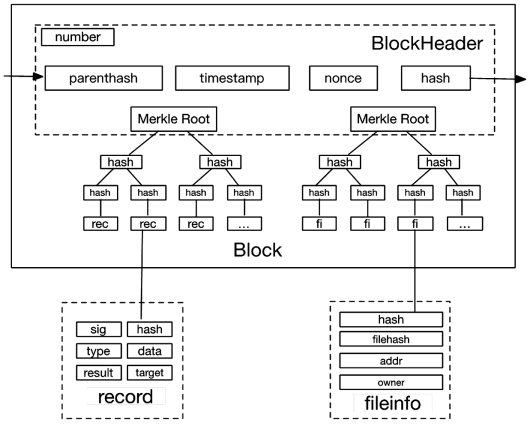

Fig. 3. Data structure for auditing

Different from blockchain in electronic currency, our system changes the original transaction structure to record user's operations on file. As shown in Fig. 3, the record structure retains the fields of user's signature (sig) and the hash in the original translation structure. The field of type is added to specify the user's operation type, and the field of result is used to store the hash value of data after being operated by users. Data field can't be empty if and only if the user manipulates new data. The encoded data will be stored in the data field, and the other operation types are empty. The field of target holds the hash address of data and is used to locate the data. In addition, we have added an array in the block body to save the file metadata information. Correspondingly, a Merkle root hash is added to the block header to verify the integrity of file information.

The fileinfo field is used to save the metadata information of user data. It contains the data owner, the filehash which is used to verify data integrity, the actual storage address of the data, and the hash of the fileinfo structure.

3.3 Auditing Description

Most of traditional behavior audit scheme uses logs to record the behavior, the log data is not only numerous but also easy to be modified. In this system, we build two Merkel trees in the block body. One is used to save user behavior records, and the other is used to save metadata information of a file. If the block data is tampered, an error will be discovered while verifying the block information and then the block will be rejected.

***Data Source Audit*:** One of the core elements of audit is to ensure the accuracy of the data source. If the data source is not trustworthy, the result of auditing is suspect. In this system, any operation on the file will automatically generate a corresponding operation record in client. These operation records are signed by user with their private key and then broadcasted to the blockchain network. The miner nodes in the blockchain layer accept record. They first verify the record integrity and then pack the valid record into a block, thus ensuring the security of data source.

***Integrity Audit*:** The files' metadata stored in the block body can be used for integrity auditing. The hash value of file has been recorded when upload the file. In other words, the initial state of a file has been recorded. When the file is obtained through the corresponding URL, its hash value is recalculated. If it is not equal to the hash value stored in the block body, the file is considered to be damaged.

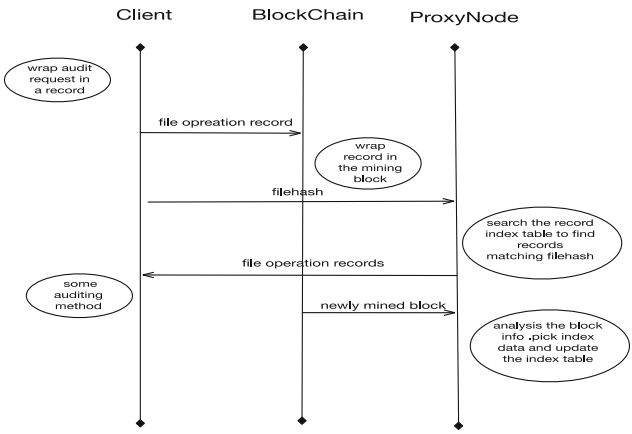

Fig. 4. The process of behavior auditing

***Behavior Audit*:** Figure 4 displays the process of behavior auditing. The file operation record stored in the block body can be used for behavior auditing. Since the audit data is stored in the block and the number of block is gradually increasing with the time,

acquiring the block data directly will be a time-consuming operation. Therefore, a proxy node is added to our system for keeping the index of block data to accelerate the locating block. When the block is dug out by miner, it will be broadcasted to all nodes in the network. The proxy node will perform preliminary analysis, it stores the user's identification information into the file table, and saves the operation record into the record table. When we want to locate the file, we need first access to proxy node's file table, query the relevant records to find out which block the metadata is stored, and then we get the corresponding block from the block chain to take file metadata information. For the audit request, we also send a request to the proxy node so that we can find out the location of block where all historical records are stored, and then achieve the corresponding record information for subsequent audit and analysis operations.

3.4 Security Analysis

In this section, we analyze the security of our scheme through two common attacks.

***Repudiation Attacks*:** The records are stored in block chain, and the copies are stored in various nodes in blockchain network. As the record chain is growing all the time, modifying some nodes' copies can not repudiate what he has done.

***Replay Attacks*:** Every time a record is broadcasted to the miner, miner will check the hash of the record which contains a property named nonce. Nonce is a unique mark to identify a record generated by the blockchain network. When user wants to replay a record, he would construct a record, however the nonce is generated automatically, so the two records are different.

4 Performance Evaluation

In this section, we measure the performance of our scheme. We developed a prototype on Ethereum [1] platform using Aliyun as data storage service and test the performance of uploading and download different size of file. Our implementation uses the PBC library at version 0.5.14, OpenSSL library at version 1.0.2n. We choose AES-128 for block encryption and decryption, SHA-1 for hashing, RSA-1024 for verification. All experiment results are on the average of 20 trials with the top and bottom results excluded.

We prepared some files which size varies from 2 MB to 512 MB for this test. We record the time of uploading file, broadcasting operation information and packing information into a block. Then we download the uploaded files and record the downloading time. By comparing with a general storage system, we analyzed the time overhead of this system.

4.1 Time Overhead of Uploading File

The uploading process is divided into three steps. First, the user uploads files to cloud storage server, then the user broadcasts the request of storing files to the network,

finally, the user waits for the system to pack this record into a block. The percentages of the time for uploading, broadcasting, and packing into blocks are shown in Fig. 5.

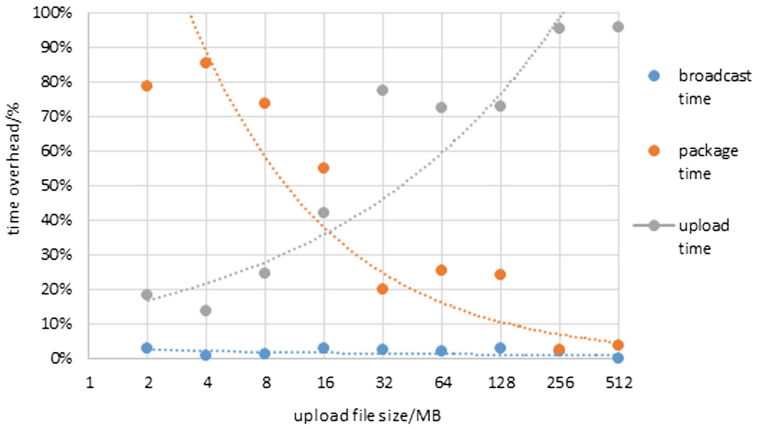

Fig. 5. Time overhead for upload download file

From the Fig. 5, we can see that the broadcast operation occupies a small time in the entire file upload process, and gradually decreases as the data increases. After adding the waiting time of packing into a block, we can see that the time delay has a remarkable rise when the uploading file data is small, and the percentage of time spent packing into a block in the entire operation is also high. However, as the size of data increases, the time allocation of packing into a block in the total operating time shows a decreasing trend. When the file size is up to 512 MB, the time allocation is even lower than 2%.

From the Fig. 5, we can see the waiting time of packing into a block is not fixed, for that mining is a process of calculating random numbers, so there may be some cases where the speed of mining is too fast or too slow for some test cases. However, on the whole, when the file is small, the time spent waiting for packing into a block takes a high proportion. When the file is large, the network I/O, that is, the upload time takes a high proportion.

4.2 Time Overhead of Downloading File

The downloading process is also divided into three steps. First, the user accesses the agent node to obtain the metadata information of the file, then the user could obtain the file from cloud storage server according to the file URL, and finally broadcasts this operation to the blockchain network waiting for the network to pack this operation into blocks. The percentage of total time for these three periods is shown in the Fig. 6.

As can be seen from the Fig. 6, obtaining the file's metadata information only exists in the process of interacting with the proxy node, and the time-consuming tends to be stable. The downloading time increases as the size of the file increases, and the proportion of the downloading time to the total downloading time is also increasing,

however, the proportion of time spent broadcasting and packing into a block to the total downloading time is decreasing. This shows that with the increase of files' sizes, the impact of the network's I/O on the system is greater than that of waiting for packing into a block.

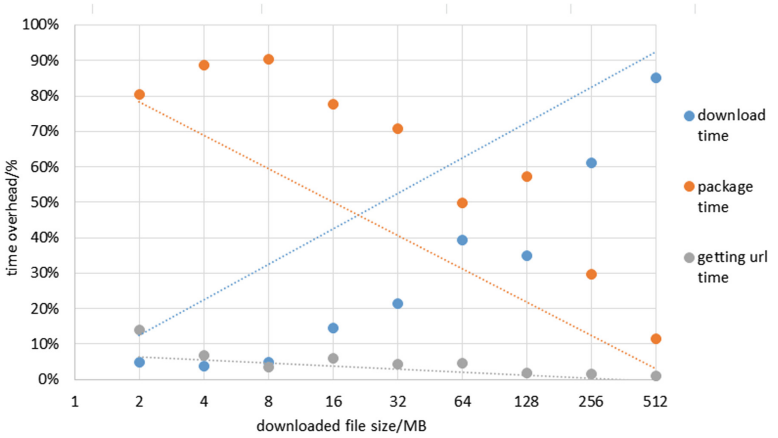

Fig. 6. Time overhead for download file

5 Conclusions

This paper builds a blockchain-based behavior audit framework that uses blockchain to store files' metadata information and users' behavior information. The framework implements operations such as auditing the integrity of files and auditing users' behaviors. Compared with the traditional logging-based audit method, the security of the audited data is guaranteed. Although the proxy node is used to speed up the query of operations on the block, due to the problem of packing delay in the blockchain system, the file records may be packed into the block for a long time, resulting in a long waiting time for the user to confirm that the operation is recorded in the log. In the meantime, it takes a long time waiting for packing into a block when files are stored, which may lead to that users successfully upload files but cannot immediately query their own files. Through the test we have found that when the file size is increased, the total time spent on packing records into the block gradually decreases.

Acknowledgments. This work is supported by the National Key R&D Program of China (2016YFB0800402), partially supported by the National Natural Science Foundation of China under Grant No. 61232004 and the Fundamental Research Funds for the Central Universities (2016YXMS020).

References

1. G Wood Ethereum: a secure decentralised generalised transaction. http://www.ethereum.Org
2. Dong, C., Wang, Y., Aldweesh, A., et al.: Betrayal, Distrust, and Rationality: Smart Counter-Collusion Contracts for Verifiable Cloud Computing, ACM CCS. ACM, New York (2017)
3. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Future Gen. Comput. Syst. (2017)
4. Oliner, A., Stearley, J.: What supercomputers say: a study of five system logs. In: IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 575–584. IEEE Computer Society (2007)
5. Itani, W., Kayssi, A., Chehab, A.: Privacy as a service: privacy-aware data storage and processing in cloud computing architectures. In: The 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC (2009)
6. Ateniese, G., Burns, R., Curtmola, R., et al.: Provable data possession at untrusted stores. In: ACM Conference on Computer and Communications Security, pp. 598–609. ACM (2007)
7. Tian, H., Chen, Z., Chang, C.C., et al.: Enabling public auditability for operation behaviors in cloud storage. Soft. Comput. **21**(8), 1–13 (2016)
8. Nakamoto, S.: Bitcoin: a peer-to-peer electronic ash system. Technical report (2009). https://bitcoin.org/bitcoin.pdf
9. Sengupta, B., Bag, S., Ruj, S., et al.: Retricoin: bitcoin based on compact proofs of retrievability. In: The 17th International Conference on Distributed Computing and Networking (2016)
10. Ramachandran, A., Kantarcioglu, D.: Using Blockchain and smart contracts for secure data provenance management (2017)
11. Yang, C., Chen, X., Xiang, Y.: Blockchain-based publicly verifiable data deletion scheme for cloud storage. J. Netw. Comput. Appl. **103** (2017)
12. Dagher, G.G., Mohler, J., Milojkovic, M., et al.: Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology, sustainable cities & society (2018)
13. Ghoshal, S., Paul, G.: Exploiting block-chain data structure for auditorless auditing on cloud data. In: Ray, I., Gaur, M.S., Conti, M., Sanghi, D., Kamakoti, V. (eds.) ICISS 2016. LNCS, vol. 10063, pp. 359–371. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49806-5_19
14. Fu, Y.: Meta-key: a secure data-sharing protocol under blockchain-based decentralised storage architecture (2017)
15. Merkle, R.C.: Protocols for public key cryptosystems. In: Proceedings of IEEE Symposium on Security and Privacy (1980)