Visual Blockchain Using Merkle Tree

Rui Hu

A thesis submitted to the Auckland University of Technology
in partial fulfilment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2019

School of Engineering, Computer and Mathematical Sciences

Abstract

Although our communities are paying extensive attention to the blockchain technology, it is still far away from real applications. Due to the combinations of multi-disciplinary technologies in blockchain, the popularization of this technology is still at its infancy stage. Thus, we have the opportunity to understand this red-hot technology and attempt to employ it for visual applications.

To bear this in mind, we focus on online videos that collect sufficient user clicks owning to the high demand every day. When we watch the videos of TV drama episodes on an online website, we often need to organize the playlists in an ascending or descending order accurately. However, video websites such as YouTube could not provide this service due to multiple reasons.

In this thesis, we created a private blockchain for these video websites and applied the Merkle tree to store the sorted videos in the chain. Finally, a sorted playlist has been created on the video website.

To our knowledge, this is the first time to combine blockchain and online videos, which sort online videos in the right order in a playlist automatically. Our contribution is to get out of the box of video search and provide a quick video ranking solution, ultimately save the browsing time of users. Our sorting results are evaluated by using edit distance.

Keywords: Video data, blockchain, elliptic curve digital signature algorithm, Merkle tree, visual blockchain, distributed storage system

Table of Contents

List of Figures

List of Table

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature:  Date: 20 February 2019

Acknowledgment

This thesis was finished as the part of the Master of Computer and Information Sciences (MCIS) course at the School of Engineering, Computer and Mathematical Sciences (SECMS) in the Faculty of Design and Creative Technologies (DCT) at the Auckland University of Technology (AUT) in New Zealand.

First of all, I would like to thank my family, I cannot finish my study without their support and help. Next, I scincerely appreciate my primary supervisor Dr Wei Qi Yan. Under his guidance, not only the plan of this thesis was determined, but also the overall framework was clearified. Additionally, he always concerns with my process and provides invaulable help for my writing, thus this thesis could be completed in time.

I also should say thanks to my secondary supervisor Dr William Liu; I am grateful for the school administrator of AUT for their support and concerns in this thesis year as well.

Rui Hu

Auckland, New Zealand

22 February 2019

Chapter 1
Introduction

The first chapter is constituted of five parts: in the first part, the background and motivations of the related technology of this thesis will be introduced; network-based blockchain will be created to store big data like videos. The details of the blockchain will be briefed as well. To solve the problem of sorting related videos on an online video website and ensure the permanent preservation of the videos, a visual blockchain-based online video website was created. In the fourth part, our objectives will be justified. Furthermore, the contributions will be followed. We will present an overview of the structure of this thesis at the end part of this thesis.

1.1 Background and Motivation

With popularity of the Internet, it has brought great convenience to us, especially eliminating geographical restrictions and improving efficiency. People have gradually formed a network-based life circle (Li, Song, Mei, Li, Cheng, & Sun, 2018; Neyer & Geva, 2017). People's reliance on the virtual world is growing very fast; the data from GlobalWebIndex shows that users spent an average of six hours on the Internet per day. Meanwhile, at least half of the time is related to watch online videos. However, in some stream websites, we have found that the videos have the below issues concerned (Wang, Zhang, & Zhang, 2018).

First, they will question the authenticity of these videos, wonder the falsification of the visual data. Second, if they would like to play the same video, usually they need to repeat the search operation for multiple times to get the footage watched completely. The last one is that while playing the current video, if there is no video episode to be played in the playlist, or if we want to go back to the previous chapter, it will waste us much time to get the video using the current search operation.

To resolve this problem, we would like to introduce blockchain and the utilization of the chaining function to link all videos together so that when we get one video on the chain, other videos could be automatically found based on the chain.

Blockchain is an epoch-making technology and famous for its decentralization, traceability, durability, and robustness. This technology can resolve the crisis of trust well, and it has attracted eyeballs of many companies for constructing a reliable financial and credit reporting systems (Zhou, Wang, Sun, Lv, 2018; Aste, Tasca, & Di Matteo, 2017; O'Dair & Beaven, 2017).

Meanwhile, blockchain technology has a time stamp, and this feature can be applied to sort the content and information recorded on the blockchain in the right order (Liu & Zou, 2018). Theoretically combined with blockchain technology to online video website technology, it can be applied to solve our online video watching problem effectively. Combining blockchain and visual information, we call this new technology as a *visual blockchain*.

Videos needed to be stored in a blockchain could be arranged automatically and orderly, they are easily to be examined after writing them into the chain, and hard to be modified. Traditional data structure is hardly operated without centralization administration; it is challenging to implement the data contents written into the library completely automatically and arranged in a specific order since we select blockchain as the storage to store videos rather than a general data structure (Herlihy, 2019).

However, since blockchain has been created mainly for payment and transaction of digital currency, each block in the chain has only 1MB space (currently expanded to 2MB); therefore, the scope for visual blockchain is minimal. With the development of this new technology (Chen, Ding, Zheng, Yang, & Xu, 2018), the demand for preserving digital objects in the upper chain is soaring. Thus, one of the critical issues in this thesis project is how to store the big visual data in this small blockchain space.

Due to the limited capacity of the current blockchain and the high price of the cryptocurrency, the way to save data in the blockchain by using the transactions is to waste not only our time, but also our costs (Shiyong, Jinsong, Yiming, & Xiaodi, 2017).

Currently, a myriad of mainstream solutions for blockchain expansion is available. There are two standpoints: one is the expansion of Layer 1, which is called the bottom layer expansion. The plan is to increase capacity of the underlying blockchain itself. The second is the expansion of Layer 2, we use the chain structure and put a large part of blockchain into another to get a much big space (Ren, Liu, Ji, angaiah, & Wang, 2018).

In this thesis, we create a private chain based on a local server, then utilize the smart contract and Merkle Tree to save big visual data in the chain. Thus, we solve this problem by not saving the big data of videos on the chain to achieve visual blockchain and finally apply it to sort the sequence of the online videos.

1.2 Research Questions

The focus of this thesis is on how to save big data on a blockchain. There are still many technical problems and barriers in the current mainstream expansion. Due to the needs of technological development, the requirement of big data storing in the block will lead to

capacity expansion of each block. At the same time, due to the working principle of blockchain, each block is generated by miners who record the transaction in the ledger, and these blocks are linked as a chain one by one in this way. When the block capacity expands, more data needs to be registered into one block, which quickly leads to delay in transactions (Manski, 2017).

The price of expansion will lead to delayed transactions, and fast transaction reduces the storage of each block. Therefore, the research questions in this thesis are:

(1) Are there any ways to save big data into the blockchain without affecting the transaction speed and transaction efficiency of the entire blockchain?

(2) Is there any method that can help online video websites arrange their playlist?

This thesis will solve these two problems. First of all, the invention of blockchain itself is not intended to be applied as a database, so it does not have the inherent advantages of data preservation and data extraction. Secondly, blockchain technology is essentially a peer-to-peer network, cryptographic signature, data storage, distributed product, and a collection of various underlying technologies (Li, Wu, & Chen, 2018). We need to extend and refine the two questions. We can think the issues from another perspective: what is the purpose of saving big data to the chain?

It is to ensure the authenticity of the visual data and is not easily to be tampered with; when the data is needed, it can be extracted and self-certified that the information is the original. Second, merely expanding the capacity of each block on the blockchain will inevitably affect the efficiency of mining and indirectly influence the effectiveness of the transaction. Attempting to implement big data uplink storage through a solution does not seem to apply.

In summary, the problem is refined. The best way to solve this problem is to use the storage under the chain and guarantee its authenticity, reliability, and immutability.

1.3 Contributions

The main contributions of this thesis include:

(1) Solving the problem of the big data storage of the current blockchain technology

(2) Realizing the safe and reliable preservation of original data

(3) Listing videos of online video sites in an orderly arrangement.

This thesis is based on blockchain technology, which has the characteristics of decentralization and is hard to be rewriten; meanwhile, it can guarantee the authenticity of the data stored in the chain. Besides, we commit to investigate the big data stored on the chain without affecting the transaction speed of the blockchain.

1.4 Objectives of This Thesis

In this thesis, we created a private chain to achieve the preservation of big data. At the same time, it is inspired by the principle of Merkle tree adopted by using blockchain itself. The principle of a Merkle tree is combined with the smart contract to record big data into the blockchain, permanently and reliably. Therefore, the main objectives of this thesis are:

(1) The idea of the entire project

(2) Specific algorithm and principle applications of Merkle tree

(3) Construction of local private chain;

(4) Storing the online videos in the blockchain, and using the order in the blockchain to achieve the order of the playlists.

1.5 Structure of This Thesis

The structure of this thesis is as follows. Chapter 2 includes literature reviews related to previous work in blockchain, ETH, Merkle tree, and so on. Thus, in Chapter 2, we will introduce background knowledge about blockchain and Merkle tree, which provide a theoretical basis for all relevant applications. In Chapter 3, we will discuss research methods, including the design of entire experiment as well as its methods and expectations. In Chapter 4, we will implement this test and compare the outcomes with the experimental expectations. In this chapter, we will present our experimental results; additionally, the limitations of this design will be listed out. In Chapter 5, the results of

the experiment will be discussed and analysed. In the last Chapter, our conclusion will finally be drawn, and the future work will be given as well.

Chapter 2
Literature Review

In this chapter, we will devote to in-depth study and analysis of current developments through extensive reading and access relevant literature. This chapter will focus on the relevant knowledge of the Merkle tree, the distributed storage of blockchain, and the combinations of them. Finally, the practical value of big data uplink will be surveyed.

2.1 Introduction

The Internet is growing virally, typically people have paid much attention to blockchain technology and its applications. A revolution triggered by blockchain technology is quietly coming. The use and popularization of the Internet have updated the way of information distribution on our daily social lives (Yuan & Wang, 2018).

The emergence of blockchain technology not only creates new digital currency but also has a significant impact on our financial systems. In this era, it has epoch-making significance for reshaping the bank credit systems. Therefore, the applications and popularizations of blockchain technology for today's socio-economic pattern are out of our imagination (Hawlitschek, Notheisen, & Teubner, 2018).

Since blockchain technology has developed for a short period, it is still continuously developed like other technologies; the topmost representative of blockchain application is Bitcoin, which has a myriad of advantages. However, it also accompanies with apparent problems.

2.1.1 Mining Waste Resources

As early in 2014, some people had conducted statistics, the total network computing power was 110 million GH/s. According to this speed, the power cost is about 70.712 million dollars, and the equipment investment is about 0.733 billion dollars. The mine cost is about 0.8 billion dollars, and it shows that the waste is enormous.

Meanwhile, the machine investment for bitcoin mining is vast. Therefore, it estimates that this calculation requires about 36,670 KnCMiner Neptune mining machines, each with a power of 3,000 GH and a price of 9,995 USD considering the problem of loss here. According to the frequency of two updates annually, the result is about 733 million US dollars per year; Also, these mines will also produce carbon dioxide with an exhaust gas volume of approximately 424,725 tons per year (Hass, 2014).

Of course, the purpose of mining is to make a hash collision to purchase the accounting authority to obtain virtual cryptocurrency, and it does not produce other real value; therefore, its power consumption and mining machine investment are considered a waste of resources. When considering improving this problem in other systems, the following two methods are mainly used.

The first is to eliminate the single pow (proof of work) mechanism and introduce the pos (proof of stake) mechanism, the pow collaborative maintenance blockchain system is stable and reliable (Balajee, 2018). The proof of equity is a kind of currency ownership. In blockchain, the system determines accounting authority according to the proportion and possession time of each node (Fairley, 2017). The prover needs to provide a certain amount of cryptocurrency. Ownership plays the pivotal role for system evaluations.

Second, as many nodes in the whole network compete for the billing rights of each block, which leads to the calculation of the entire network. The difficulty of hash collision is also climbed as an exponential increase to reduce such problems (Subramanian, 2018). Some systems applied private chain or alliance chain to directly assign the billing rights to specific nodes instead of using the traditional method of competing to compete for billing rights (Tang, Shi, & Dong, 2019).

2.1.2 Block Capacity and Transaction Speed Limit

When Bitcoin was designed, it set a capacity of 1MB for each block so that it can accommodate 4096 transactions. At the same time, the workload proof mechanism makes it necessary to confirm the transaction and records the transaction on the blockchain (Zhao, Wang, Li, & Li, 2018). Currently, the Bitcoin technology is not mature enough as the major credit card networks; the work to increase this limit is still underway.

Besides, block expansion has become urgent. Bitcoin developers have proposed a switch from Bitcoin Core to Bitcoin XT, which was expanded from 1MB to 8MB and will be doubled every two years in future. This change requires 750 of the 1000 contiguous blocks to get miner's approval. This incident is currently under development (Wang, & Kogan, 2018).

2.1.3 Lack of Turing completeness

As blockchain can ensure that Bitcoin transactions are not falsified, it is theoretically guaranteed that any code that has been written cannot be counterfeited. However, Bitcoin's scripting language is not Turing complete, it does not support looping statements; that means Bitcoin can only be used as a digital currency and cannot directly support smart contracts or more complex decentralized applications (Dinh, Liu, Zhang, Chen, Ooi, & Wang, 2018).

The blockchain platform Ethereum Virtual Machine code is Turing complete; the programming based on EVM platform theoretically can achieve any imaginable calculations, including infinite loops, etc. Ethereum enables any developers to deploy their own applications, which provides a guarantee of effective execution for them (Gouru & Vadlamani, 2018).

2.2 Current Research Progress

At present, the research trends of online videos mainly focus on the following aspects:

- Comprehensive analysis based on preferences, habits, frequency from the actors who would like to watch online videos. All statistical data is not only obtained from a single video platform, but also collected from cross platforms (Zhou, Gu, Wu, Chen, Chan, & Ho, 2018).
- Online video streaming supplies users with the service, including a sorted advertisement, game, and other contents related; these are results based on the users' behaviors (Hasan, Jha, & Liu, 2018). Meanwhile, a user who frequently browses online videos will be recommended to watch exclusive videos based on the accurate inference of the user's preferences (Tan, Guo, Chen, & Zhu, 2018).
- With 4G applied to more and more fields, online video platforms are not only limited in the PC system, they have already transferred to the mobile device system; the statistics of their energy consumption is calculated by using the changes (Zhang, Wang, Quan, Yin, Chen, & Guo, 2018).

- At present, YouTube for video indexing is mainly divided into two parts, recall and sorting. As shown in Fig. 2.1, the first green funnel is the recall algorithm, which can help a recommendation system to filter hundreds of videos from millions of objects resources (Paul, Jay, & Emre, 2016). Additionally, other recall sources can be added except the recall algorithm used with deep learning here; these resources will be transferred to the next part. Because of the large amount of computation, it is impossible and unnecessary to use all features of the recall algorithm; thus, the recall algorithm only uses user behaviour and scene characteristics. The ranking algorithm uses more features, calculates a score for each candidate video, and ranks the ratings from high to low. In this way, dozens of objects will filter from hundreds of videos. The offline indicator and the online AB test are applied to the evaluations of this algorithm, and AB test is set as the main evaluation index.

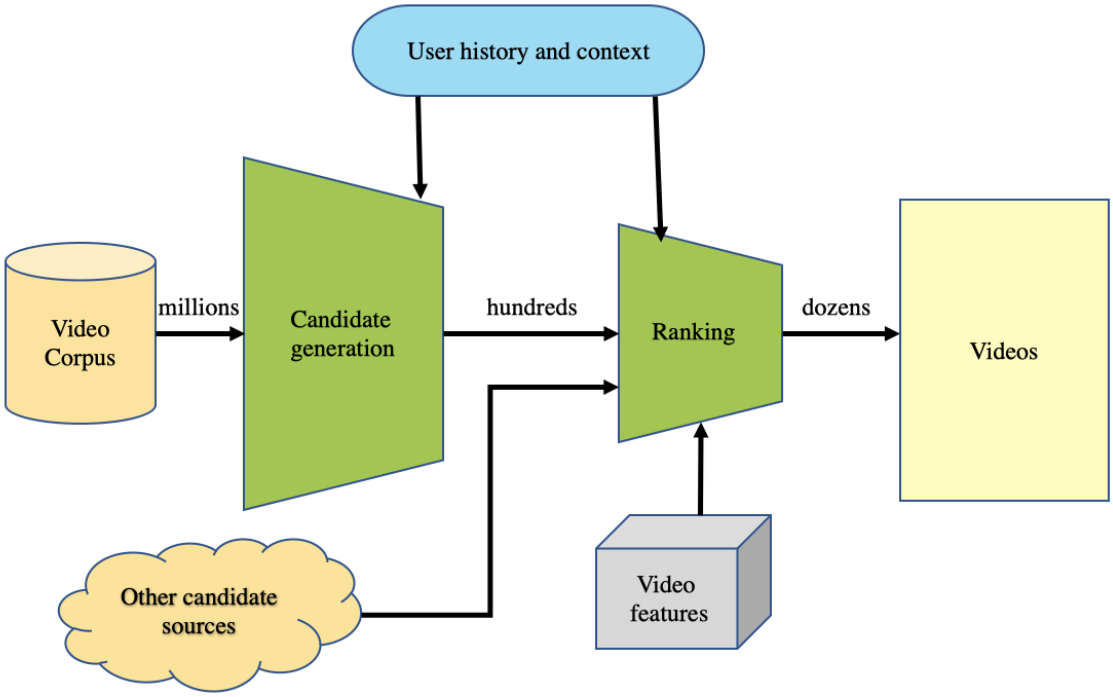

Figure 2.1 Recommendation System Architecture

The research outcomes based on blockchain show it is mainly concentrated in the following two directions:

- The development of cryptocurrency platform based on the blockchain creates such a forum to enlarge the influence of this cryptocurrency and eventually to increase

the real value of this cryptocurrency (Kugler, 2018; Genkin, Papadopoulos, & Papamanthou, 2018).

- A variety of applications related to blockchain are developed, the ranges of these applications are widely in game, finance, auction, bidding, etc. (Rosa & Rothenberg, 2018; Ma, Huang, Bi, Gao, & Wang, 2018).

We have noticed that research work related to online video streaming mainly focuses on big data analysis and sorts the users to recommend advertisements or videos. In this thesis, we combine blockchain with traditional online video systems, aim to solve the problem of disordered playlist during video playing in the online video website.

2.3 Blockchain

2.3.1 The Block

As the basic constituent unit of the blockchain, it is mainly composed of a block header and a block body, where the block header is primarily responsible for storing metadata, and the block body is for recording transaction data. The block header contains three sets of metadata:

- Connecting with the previous adjacent block and indexing the data from the hash value of the parent block. The hash value in the last header block will be permanently written into the next sub-block, it cannot be falsified with continuous characteristics between blocks (Jan & Geir,2018).
- Random number (Nonce) mining difficulty and time stamp. Because bitcoin in the network issued a total of 21 million, at the beginning of each block, it will be assigned as 50. The design is to avoid bitcoin too early to finish the excavation.

  Therefore, a random number is set, it will adjust the difficulties according to the time of generated new blocks so as to assure that the interval between every two blocks is 10 minutes. This timestamp stores the time for each transaction on the chain, and none can tamper with it. Thus, it ensures that each block is sequentially connected (Siba & Prakash, 2016).

- Ability to quickly summarize Merkel tree and its structure for all transaction data in the check block (Meloni, Madanapalli, Divakaran, Browdy, Paranthaman, Jasti, Krishna, & Kumar, 2018).

Figure 2.2 illustrates the structure of each block on the blockchain, where block 0 is the genesis block. Hence, there is no hash value in the block pointing to the block header of the previous block.

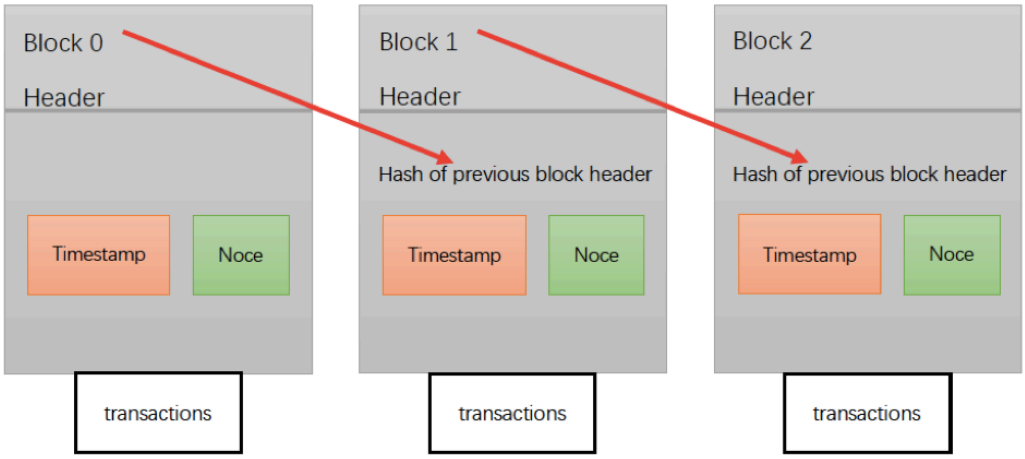

Figure 2.2 Branch of Blockchain

The first block on the blockchain was created in 2009 by Nakamoto Satoshi, known as the Genesis Zone. It is the ancestor of all the blocks on the chain, which means that we can trace back from any blocks and eventually reach the genesis block (Tsukerman, 2015). Since the genesis block was compiled into the Bitcoin client software, each node starts with a blockchain containing at least one block, which ensures that the genesis block will not be changed (Vladimir & Dmitry, 2018). Each node "knows" the hash value, structure, time of creation, and a transaction within the genesis block. Therefore, each node makes use of the block as the first block of the chain, to construct a root of a secure, trusted blockchain (Shu, Y. 2018). Figure 2.3 shows information about the first block in the Bitcoin network.

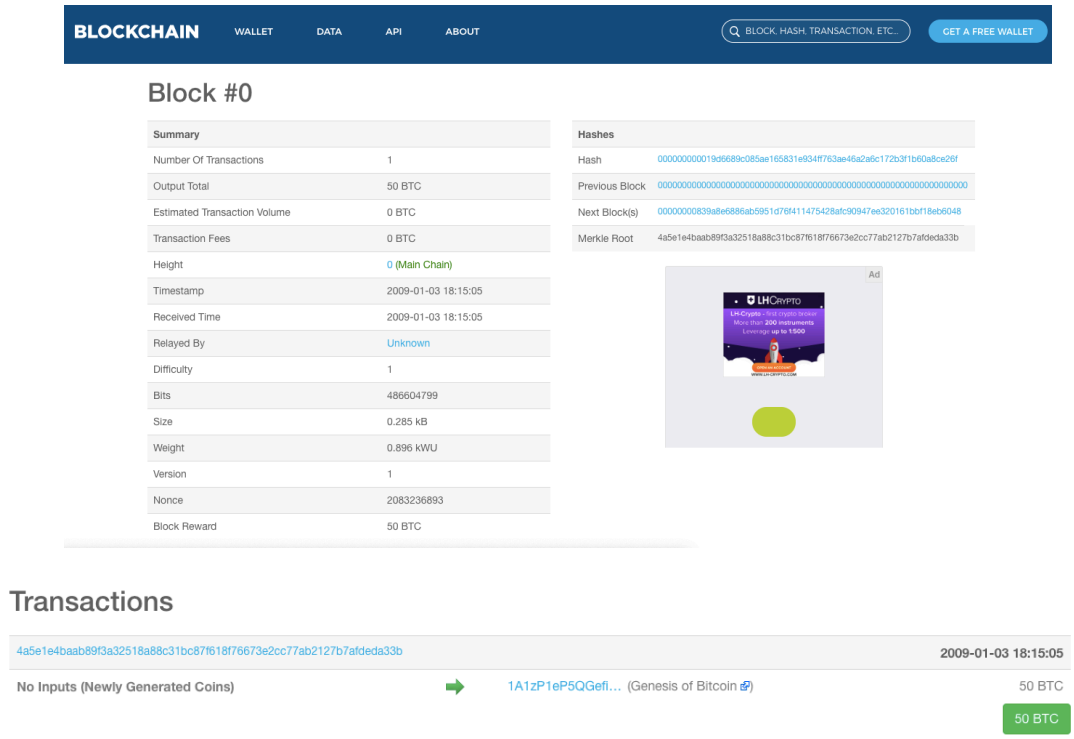

Figure 2.3 The Genesis Block

The genesis block contains a hidden message encompassing the phrase “The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.” in the coinbase transaction, which is the headline of the Times on the day (Davidson, & Block, 2015).

2.3.2 Hash Algorithm

The hash algorithm is a one-way cryptographic mechanism that guarantees that transaction information stored on the blockchain is not tampered with. Figure 2.4 describes how the hash impacts the both sides of input and output, it is obvious that each input generates one output through a hash function, and they are a one-to-one mapping. Using the hash algorithm, the received plaintext can be converted into a short length and fixed number of hash data in an irreversible way (Cho, 2018). Therefore, this algorithm has two characteristics:

- The encryption process is irreversible, which means that we cannot extract the encrypted original text by outputting the hash data.

- Input the original text and the final output hash data, one-to-one correspondence, any change in the input text will directly change the output value.

On the blockchain, SHA-256 (Secure Hash Algorithm) is usually used for block encryption. The input length of this algorithm is 256 bits, and the output is a hash of a string of 32 bits random number. The blockchain encrypts transaction data in the block by using a hash algorithm and obtains a string of 32-bit letters and numbers mixed with an arbitrary three-column string and saves it to the block, which will be the unique one (Appelbaum & Stein Smith, 2018).

A flag can identify the block accurately. Any nodes in the blockchain can obtain hash value of the block through a simple hash calculation; the comparison between the hash values and the generated initially hash values (Zhang & Fan, 2018). If the match indicates the data stored in the block has not been tampered with, the data saved in the block has tampered.

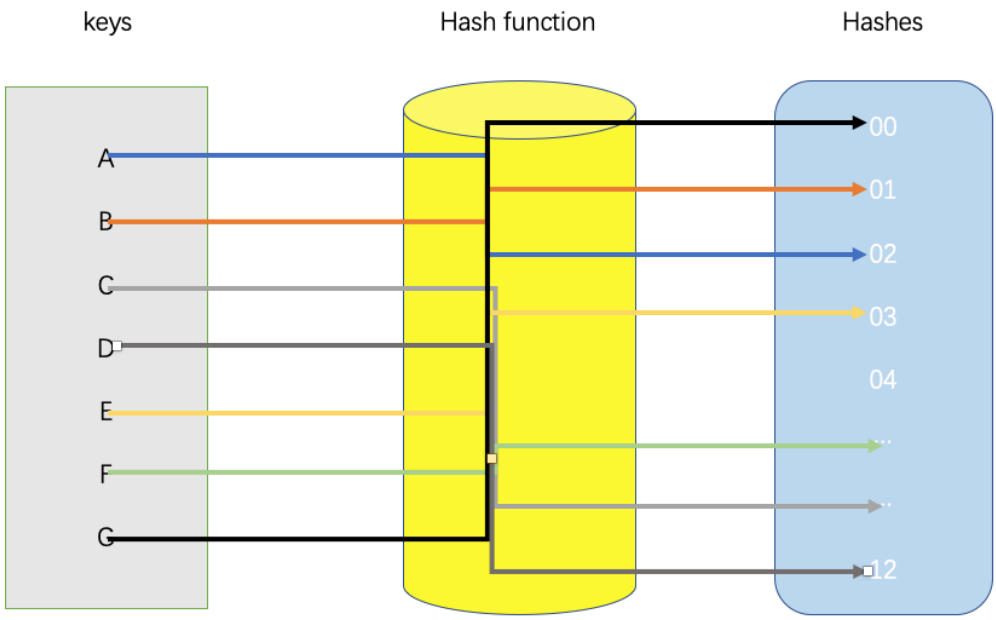

Figure 2.4 Hash Table

2.3.3 Public and Private Keys

Before discussing the problems of public and private keys, we must first understand the current symmetric encryption technology. The so-called symmetric encryption technology adopts the same secret key for encryption and decryption. The encryption is much harder than the general symmetric encryption, we call it as asymmetric encryption technology. It has been understood that encryption and decryption with different keys (Sehra, Cohen, & Arulchandran, 2018).

As shown in Fig 2.5, because the same encryption key is used for encryption and decryption by using the symmetric encryption technology, the file will be lost or stolen if it is deciphered during transmission. Seriously, it will also be associated with the key (Lacity, 2018). All documents are at risk. The asymmetric encryption technology corresponding to this is highly secure. Since various encryption keys can be applied to encryption on the encryption side, we usually utilize public key encryption and private key decryption. In the case of file transfer, if it is stolen, it will not cause all the related information to have security risks.

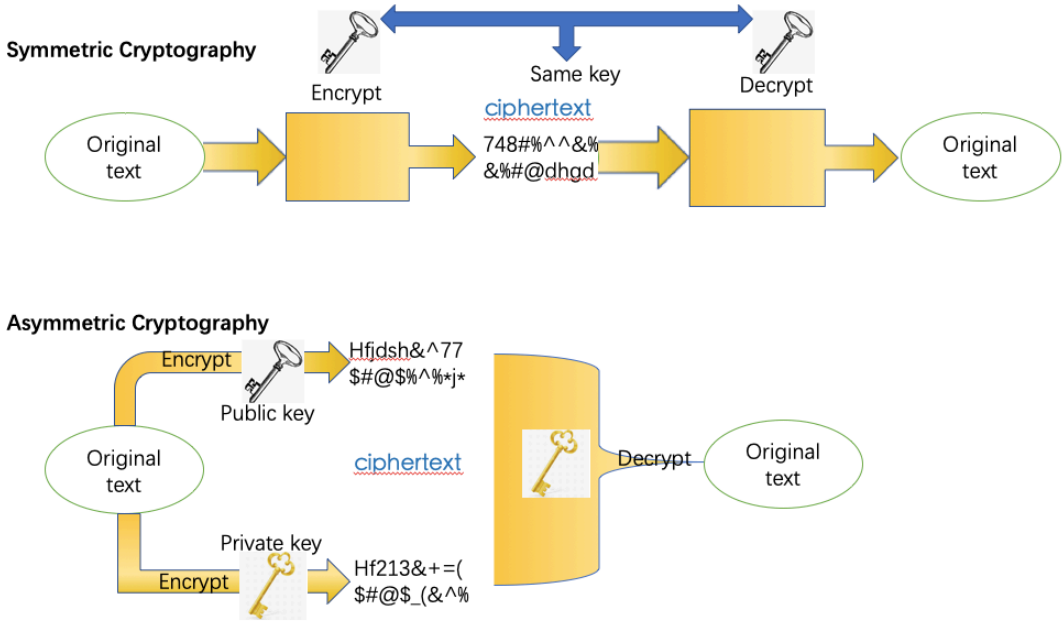

Figure 2.5 Symmetric Cryptography & Asymmetric Cryptography

In the Bitcoin system, the private key is essentially an array of 32-bit bytes that can be used to generate the public key and address. On the blockchain, the asymmetric encryption technology is generally utilized to encrypt and transmit information. As

illustrated in Fig 2.6, the sender of the message Alice (A) delivers a piece of information to the receiver Bob(B) and encrypt the data. Encryption generates a classified text, and then sends it to Bob. After received the confidential document, Bob decrypts it with its private key to obtain the original version, thereby implements the encrypted transmission of the information.

Besides, to encrypt the information using key pair, the public and private keys can also be optimized for identification as illustrated in Fig 2.6. For the sake of proving its identity to Bob, a piece of information is delivered. Since Alice's public key is openly available on the whole network, Bob can sign and verify the signed information with Alice's public key. If matched, it proves the authentication. Mismatching description of Alice shows that it is a fake. Since Alice can only hold Alice's private key, it thus can be authenticated.

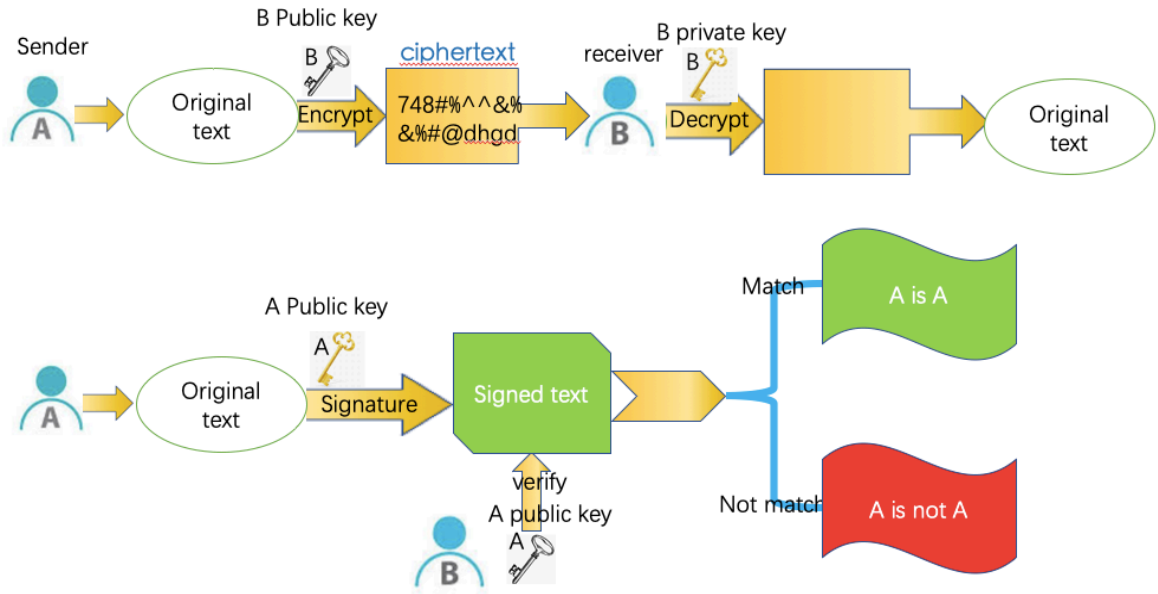

Figure 2.6 Asymmetric Cryptography Applied to the Blockchain

2.3.4 Timestamp

The timestamp on a blockchain is presented from the moment the block was generated. It records each transaction and proves the authenticity of the transaction (Stavrou & Voas, 2017). The timestamp was written directly into the blockchain, the generated block on

the blockchain cannot be modified; because once tampered, the generated hash value will be modified and become invalid. Each timestamp is counted in its random hash value. This process is repeated, and the end-to-end connection will eventually generate a complete chain. As explained in Fig 2.7, each timestamp should include the previous timestamp in its random value, and each subsequent timestamp will enhance the previous timestamp. Thus, it is hard to form a whole Tampered chain (Pierro, 2017).

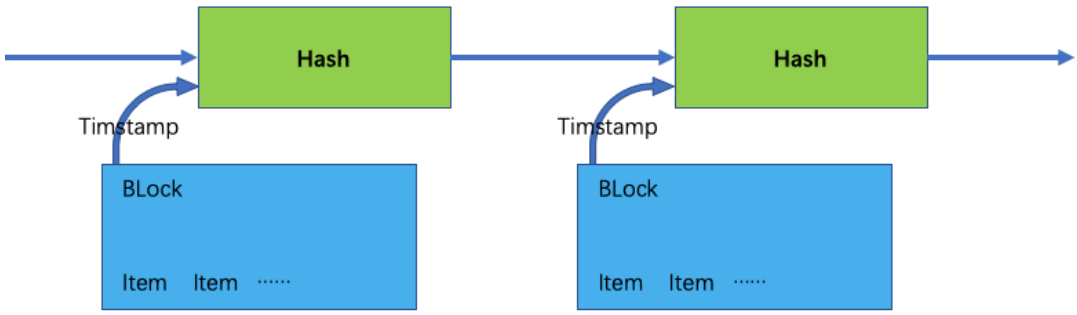

Figure 2.7 Timestamp Reinforcing

The time stamp maintains the regular operation of the entire blockchain system, assures that each miner plays an essential role in fairness. When the miners compete for the billing rights of the block, calculating the target value needs to apply the current timestamp, that is additionally recorded in the header of each block (Kokina, Mancha, & Pachamanova, 2017). Although there is no mechanism to prevent miners from using other timestamps when digging new areas, instead of current timestamp, they generally do not commit this as the timestamp verification will fail and other nodes will not accept the block, this wastes resources of the miners.

When a miner releases a new block to be dug, the other nodes will verify their timestamps and compare whether the timestamp of this new block is later than the previous one. If a miner utilizes a timestamp newer than the current, the difficulty is lower because the value is inversely proportional to the current timestamp; the network will accept the miner whose timestamp is the current one as the result of its higher difficulty. If a miner uses a timestamp that is later than that of the previous block and less than the

current timestamp, the difficulty will be higher; thus, it takes more time and effort to dig the block out. By the time once the block is taken out, there may already have more blocks so that the block will be rejected (Lu, 2018; O'Leary, 2018). In summary, miners always make full use of accurate timestamps. Otherwise, they will get nothing.

2.3.5 Merkel Tree Construction

All transactions contained in the block are firstly generated by using the Merkle tree stored in a block header. The Merkle tree algorithm was designed for synchronizing data consistency. It is constructed as a tree based on a set of hash values. The root hash value of this tree is used as a summary of the original data list. Merkle tree has the following characteristics (Ji, Zhang, Ma, Yang, & Yao, 2018):

- The data structure is a tree that can make a binary tree or multi-fork tree;
- The value of the leaf node of the Merkle Tree is the unit data of the data set or the hash value of the unit data;
- The value of the non-leaf node of the Merkle tree is the hash value of all leaf nodes.

The principle of the Merkle tree algorithm is shown in Fig 2.8.

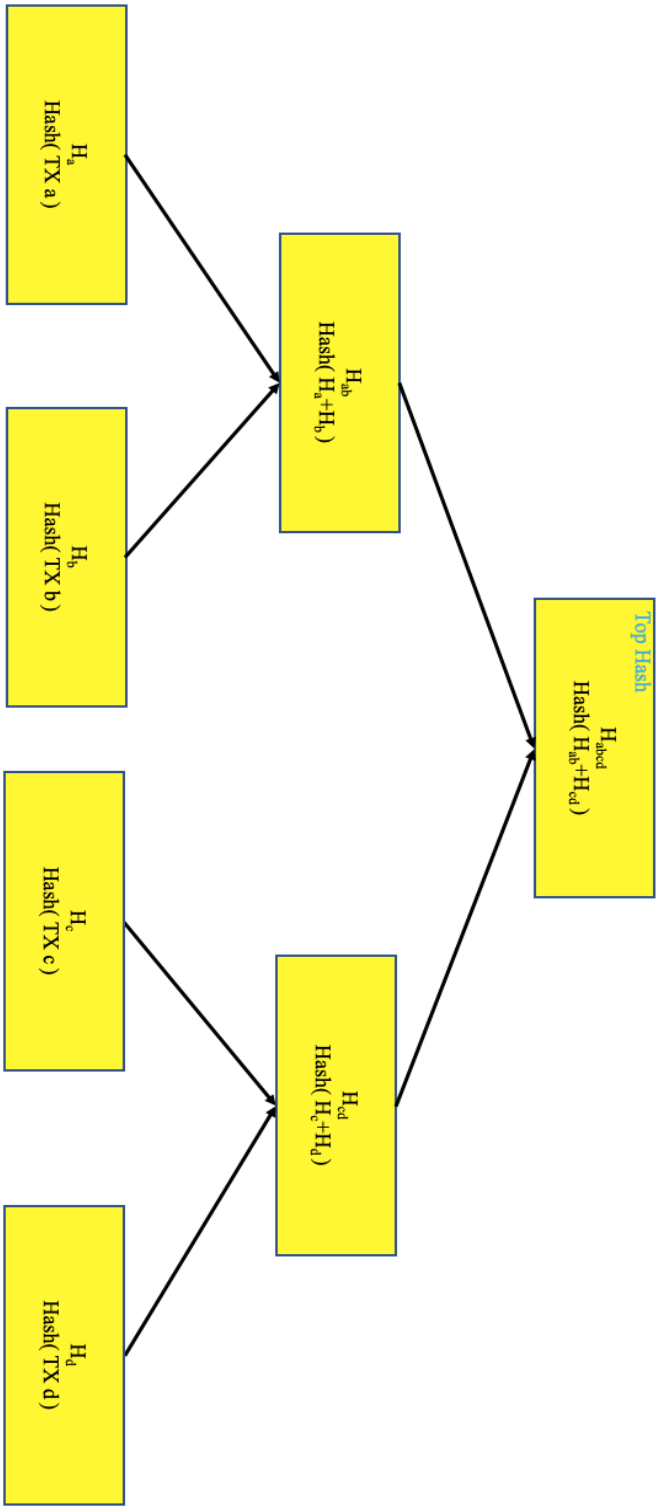

Figure 2.8 Hash Algorithm

Blockchain optimizes the data structure of the Merkle tree to store the values of all leaf nodes and generates a uniform hash based on this. The leaf nodes of the Merkle tree store the hash value of the data, while the non-leaf nodes store the hash value obtained by

combining all the leaf nodes below it. Any change of the data on the blockchain will result in the structure of Merkle tree in which the relevant data was stored (Koo, Shin, Yun, & Hur, 2018). In the process of verifying the alignment issue of transaction information, the Merkle tree can significantly reduce the amount of data calculation. After all, we only need to verify the Merkle algorithm. The unified hash generated by the Tree mechanism is excellent. The data structure of the Merkle tree on the blockchain is shown as Fig 2.9.

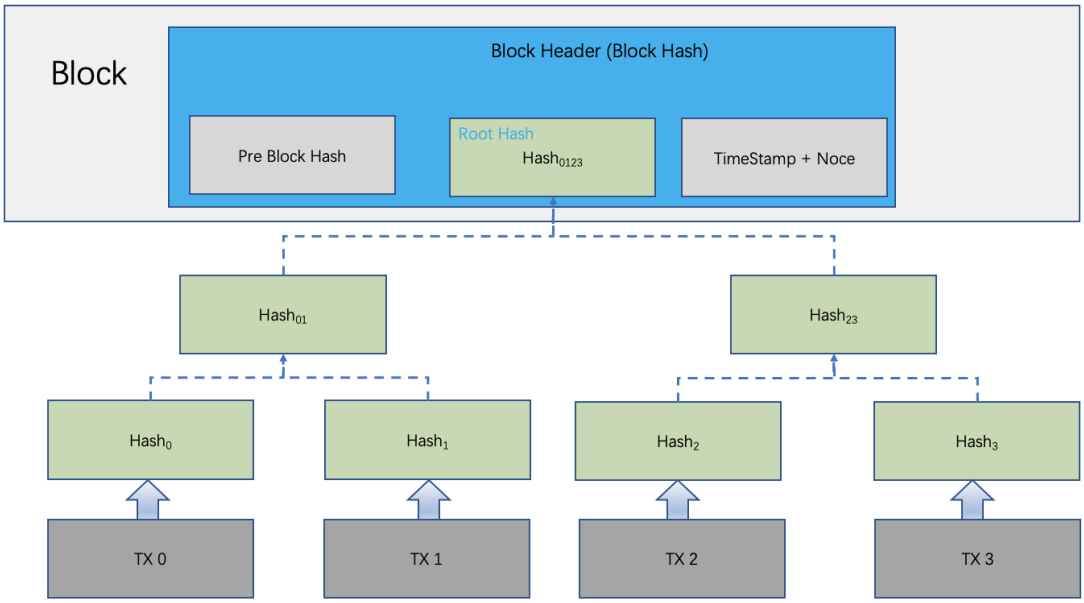

Figure 2.9 Construction of Merkle Tree on a Blockchain

2.3.6 The Infrastructure of Blockchain

The blockchain infrastructure is split into six layers: data layer, network layer, consensus layer, incentive layer, contract layer, and application layer (Thombs & Tillman, 2018). The specific structural model is indicated in Fig 2.10. The division of labour and the role of each layer are different. They cooperate seamlessly to achieve a decentralized trust mechanism.

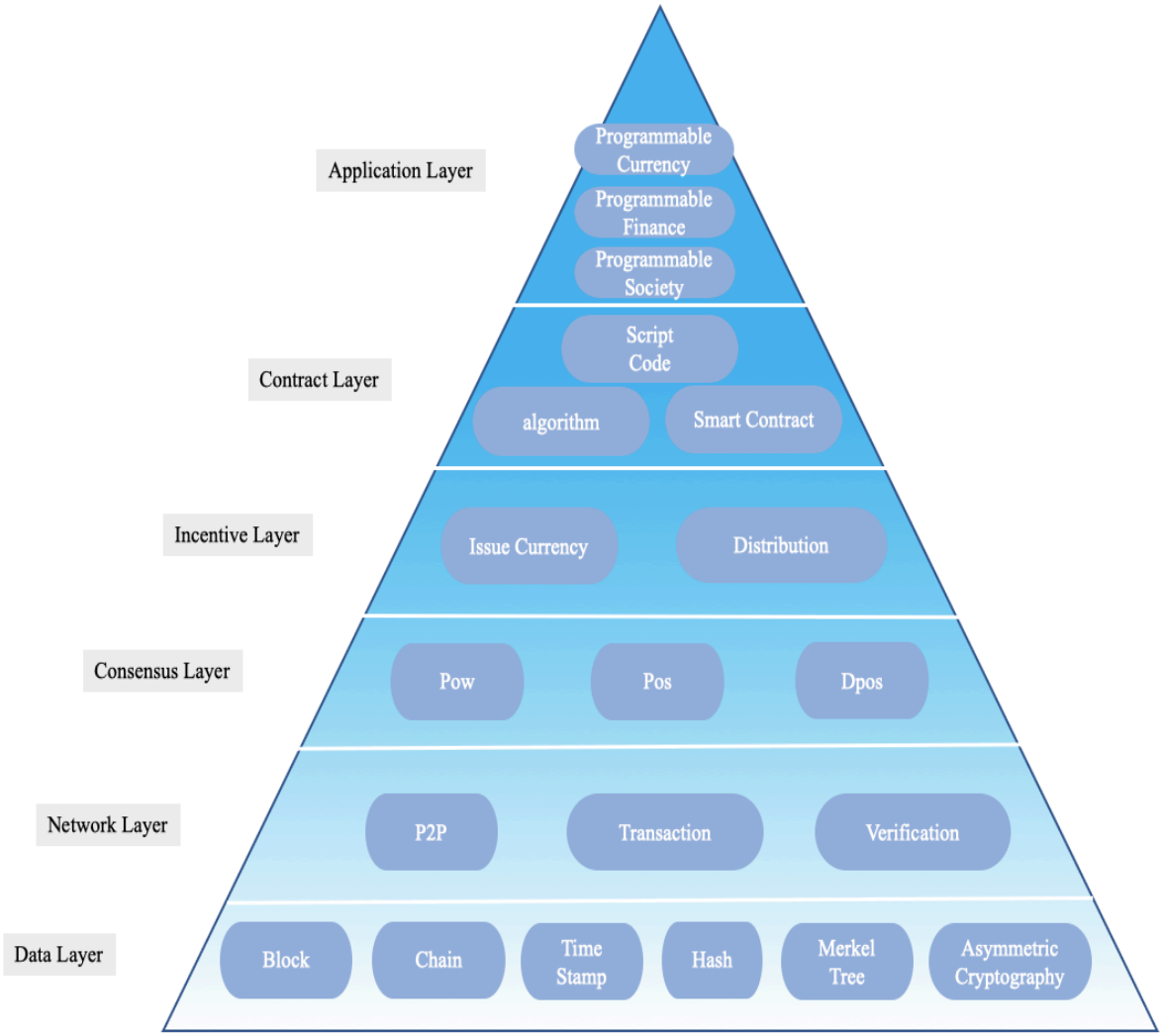

Figure 2.10 Blockchain Infrastructure

Data Layer

The data layer mainly describes the physical layer of blockchain. The first node established by the founder of blockchain is the "genesis block", then the same-sized block, created under the same rules, passes through a chain structure, the connection in turn to form the main chain (Sifah, Agyekum, Amofa, Xia, Gao, Chen, Xia, Gee, Du, & Guizani, 2018). Moreover, new blocks are continuously added into the main chain after verification; the length of this main chain is also getting longer. Each block also contains various techniques, such as timestamp technology, which enables that each block is connected in chronological order; there is a hash function that ensures transaction information is not easily tampered.

Network layer

The primary purpose of this network layer is to implement information interworking between nodes on a blockchain network. The blockchain is essentially a P2P network; each node receives information and transmits information. Nodes conserve communications by maintaining the same network (Li, 2017).

Each node in the blockchain network can create a new block. After the new block is generated, the node will be broadcasted to the whole network, and other nodes will be employed to verify the block. When the whole network exceeds 51% after the node is verified, the new blocks will be added to the main chain (Shermin, 2017).

Consensus layer

The consensus layer enables highly decentralized nodes to efficiently agree on the validity of block data in decentralized systems. The utilized consensus mechanisms on the blockchain mainly include workload proof pow, equity proof pos, and share authorization certificate Dpos. The specific working principle will be analysed on details in the consensus mechanism (Tschorsch & Scheuermann, 2016).

Incentive layer

The main function of the incentive layer is to provide particular incentives to encourage nodes participating in the security verification of the blockchain (Ao, Cruickshank, Yue, Asuquo, Anyigor Ogah, & Zhili, 2017). Let us take Bitcoin as an example. There are two reward mechanisms. Before the total amount of Bitcoin reached 21 million, there are two types of reward mechanisms. The new block generates bitcoin that is rewarded by the bitcoin (transfer fee) deducted for each transaction. When the total amount of Bitcoin reaches 21 million, the newly generated block will no longer generate bitcoin. At this time, the reward mechanism is mainly the deduction fee for each transaction.

Contract layer

The contract layer mainly refers to various script codes, algorithm mechanisms, and smart contracts. Let us take Bitcoin as an example. Bitcoin is a programmable currency. The scripts in the contract layer package specify the details of the bitcoin transaction and the details involved in the process (Risius, & Spohrer, 2017).

Application layer

The application layer encapsulates various applications and cases of the blockchain. In short, it is an application based on the combination of blockchain technology and the characteristics of various industries (Kewell, Adams, & Parry, 2017). For example, a cross-border payment transaction combines the efficient and convenient feature of blockchain technology, a supply chain tracking platform based on the traceability of blockchain technology, an investment platform and an insurance platform based on the blockchain's immutability characteristics.

2.3.7 Consensus Mechanisms

First, we must introduce the decentralization mechanism of the blockchain. Decentralization is the essential feature of the blockchain. It means that the blockchain does not depend on the management of the central node and can realize the distributed recording, storage, and update of data (Lin & Tang, 2018; Freund, 2017; Cho, Park, & Lee, 2017).

As described in Fig 2.11, in a centralized system, all users surround the central server, and all information interaction needs to pass through the central server. Not only is the interaction efficiency low, but once the central server fails, it may cause the entire network to crash.

Compared to the traditional centralization, the decentralized distribution mechanism is utilized in blockchain. The nodes are interconnected by two nodes. All nodes maintain

the entire network together. There is no central exchange server in such a network. If a node or an individual node fails, the stability of the primary network is not affected. The more nodes are accessed, the higher the stability of the entire network is.

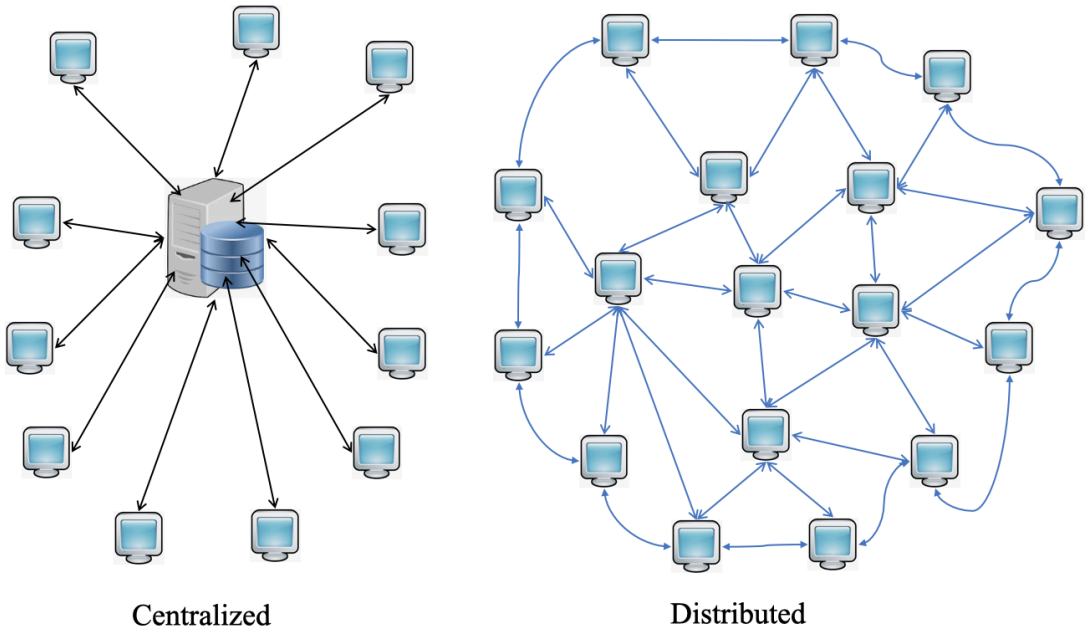

Figure 2.11 Centralized & Distributed

Next, we will introduce several consensus mechanisms used in blockchain technology and compare their advantages and disadvantages, respectively. First, let us introduce the Proof of Work (PoW) applied to the Bitcoin network. It usually only proves from the results, because the monitoring process is cumbersome and inefficient. Bitcoin employed the PoW mechanism in the block generation process. A qualified hash value for a block is composed of *N* leading zeros, the number of zeros depends on the difficulty of the network (Mahdi & Maaruf, 2018).

Getting a reasonable block hash value requires many trial calculations; the calculation time depends on the machine's hash speed. When a node provides a reasonable block hash value, it indicates that the node has undergone a plenty of trial calculations. Of course, this does not give the absolute value of the number of calculations, because finding a reasonable hash value is a probability event. When a node has a computing power of the entire network, the node has the probability of finding the block hash value.

PoW relies on machines to perform mathematical operations and obtain billing rights. It costs a large number of resources and has a high consensus mechanism. At the same time, when a consensus is reached each time, the entire network needs to participate in the calculation. The performance efficiency is relatively low, and the fault-tolerant allows 50% node error in the entire network (Alcarria, Bordel, Robles, Martín, & Manso-Callejo, 2018).

The projects applied to PoW are Bitcoin, the first three phases of Ethereum are Frontier, Homestead, Metropolis. In the fourth phase of Ethereum, serenity, proof of equity mechanism will be adopted.

The Proof of Stake (PoS) was first proposed by the "Quantum Mechanic" in 2011, and implemented in different ways by Peercoin and NXT. The main idea of PoS is that the difficulty of obtaining the node billing right is inversely proportional to the equity held by the node (Keenan, 2017). Compared to PoW, it reduces the resource consumption caused by mathematical operations to a certain extent; the performance is also improved accordingly. It is based on hash operation, and the way of competing for the bookkeeping right is weak. The fault tolerance of this consensus mechanism is as same as that of PoW. It is an upgrade of PoW. It reduces the difficulty of mining according to the proportion and time of each node's tokens, thus speeds up the process of finding random numbers (Jesus, Chicarino, de Albuquerque, & Rocha, 2018).

In PoW, a user might join the network to create a new block and get rewarded. In PoS, users can purchase the equivalent tokens and put them into the PoS mechanism as a deposit, so that the users could generate new blocks and get rewards. In general, there is a collection of money holders, and the tokens in their hands are put into the PoS mechanism, so they become verified (Gazi, Kiayias, & Russell, 2018).

For example, for the first block of blockchain, the PoS algorithm randomly selects one of the verifiers, the weight of the selected verifier is based on the number of tokens they put in, such as a verifier whose deposit is 10,000 tokens. The probability of selection is ten times more than that of a 1,000 votes verifier, given it the right to generate the next block. If the verifier does not generate a block within a specific time, the second verifier

is selected instead of generating a new block. Like PoW, PoS is based on the longest chain. With the disappearance of economies of scale (the phenomenon of increasing economic scale due to the expansion of production scale), the risk of centralization has been reduced.

The tokens are not much more rewarding, none will get a disproportionate extra return because it can afford large-scale production. All confirmations are just a probabilistic expression, not an affirmation. In theory, there may be other attacks. For example, the ETA attack of Ethereum causes the Ethereum to be hard forked; while the ETC is come along, it proved the failure of this hard fork.

BitShares community first proposed the Share Authorization Certificate (DPoS). The main difference between this mechanism and the PoS is that the node elects several agents, which are verified and booked by the agent, but its compliance supervision, performance, resource consumption, and fault tolerance with PoS (Ramezan & Leung, 2018). Similar to the board vote, the holders cast a certain number of nodes for proxy verification and accounting.

DPoS works as follows: each shareholder has a corresponding influence according to its shareholding ratio. The result of 51% shareholder voting will be irreversible and binding. The challenge is to achieve "51% approval" in a timely and efficient manner. In order to achieve this goal, each shareholder can delegate his or her voting rights to a representative. The top 100 representatives with the highest number of votes are returned to generate blocks according to the established schedule. Each representative is assigned a period to produce the block (Hu, Xiong, Huang, & Bao, 2018).

The operation mechanism of DPoS is that all representatives will receive 10% of the transaction fee, which is equivalent to an average of blocks. If an average block optimizes 100 shares as the transaction fee, one representative will receive one share as the compensation. Network delays may cause some delegates to fail to broadcast their blocks in time, which will cause the blockchain to be forked. However, this is unlikely to occur because the representative can establish a direct connection with one of the blocks before and after generating the new block.

Establishing a direct connection to our representative (and perhaps the next one) is to make sure we get paid. The DPoS voting mode can generate a new block every 30 seconds; under normal network conditions, the possibility of blockchain bifurcation is extremely tiny, even if it occurs, it can be resolved in a few minutes. The necessary steps to perform this mode are as follows:

- *Become a representative*. To be a representative, we must register our public key on the web and get a 32-bit unique identifier. This identifier will be referenced by the "header" of each transaction data.
- *Authorized voting.* Each wallet has a parameter settings in which the user can select one or more representatives and rank them. Once set, each transaction made by the user will transfer the ballot from "input representative" to "output representative." In general, users will not create transactions that are specifically for voting purposes, as it would cost them a transaction fee. However, in an emergency, some users may find it worthwhile to change their vote by paying a more positive way.
- *Keep our representative honest*. Each wallet will display a status indicator to let the user know how their representative is being performed. If they miss too many blocks, the system will recommend the user to replace a new one. If any representative is found to have issued an invalid block, then all standard wallets will ask for a new representative before each wallet makes more trades.
- *Resist the attack*. In the resistance attack, the power of top 100 representatives is the same; that means each delegate has an equal voting right; therefore, it is impossible to concentrate power on a single representative by obtaining more than 1% of the votes. Since there are only 100 representatives, it is not difficult to imagine that an attacker can perform a denial of service attack on behalf of each of his representatives in the production block.

Fortunately, since the identity of each delegate has its public key rather than an IP address, the threat of this attack can easily be mitigated. This will make it more challenging to determine the DDoS (Distributed Denial of Service) attack target. The

potential connections between the representatives will make it more difficult to hinder them from producing blocks. The comparison among them is illustrated in Table 2.1.

Table 2.1 Comparison among PoW, Pos, and DPoS

Comparison among PoW & PoS & DPoS		
Item	**Advantages**	**Disadvantages**
PoW	Simple implementation Safe and reliable Low network resource consumption	Mining causes lots of waste of resources Consensus term become longer High probability of forks
PoS	Low resource consumption	It is complicated to implement Many intermediate steps, easy to leave Security vulnerabilities High transmit pressure in network Through mining to achieve commercial applications
DPoS	Low resource consumption Low network resource consumption Short consensus term	It is complicated to implement Many intermediate steps, easy to leave Security vulnerabilities Through mining to achieve commercial applications

2.3.8 The Longest Chain Mechanisms

The global blockchain network requires all nodes to follow a formula; that is, all the blockchains saved to the local must be the longest chain, the local node verifies that. Since each block of the blockchain must refer to its previous block, the longest chain is the most difficult one to be overthrown, in order to ensure the longest chain, the miner can start calculating the next block based on any blocks (Sompolinsky & Zohar, 2018).

However, only the blocks on the longest blockchain can be systematically recognized and rewarded with mining. The rewards obtained by the packaged block can only be used after being added to the block. If we are a miner, we have dug up a new block; we get a new bitcoin reward only after we have created 99 blocks in future, we can apply the rewards in this block. It is an important mechanism to assure that the blockchain is not split shown as Fig 2.12.

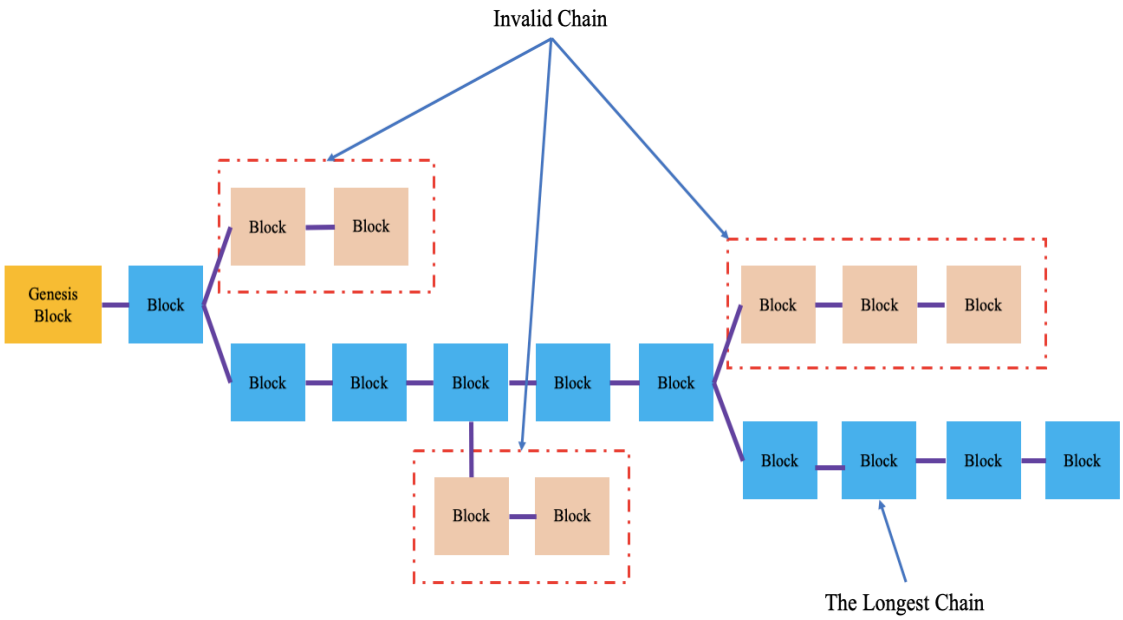

Figure 2.12 The Longest Chain Mechanisms

2.4 Merkle Tree

2.4.1 Definition

The structure of MT is shown in Fig 2.13, all the data becomes the leaf nodes of the MT throughout using the hash, each pair of the two nodes are merged into a new node through hashing again and formed as a single hash object step by step (Badra & Borghol, 2018).

The MT0123 is the root node of this Merkle tree. The leaf node is the essential element of this tree. Each leaf node contains the value after the data is hashed. Every two nodes of a leaf will generate a new child node after hashing. This child node includes the hash values of these two leaf nodes, and the resulting unique root node contains the hash value of its child nodes.

Merkle tree is also called hash tree, as the name suggests, which is a data structure that stores all hash values (Wang, Shen, Wang, Cao, & Jiang, 2018). The leaves of a Merkle tree are the hash values of data blocks (for example, a collection of files or files). A non-leaf node is a hash of its corresponding string concatenated from child nodes. In the following, we will use MT as the abbreviation of the Merkle tree.

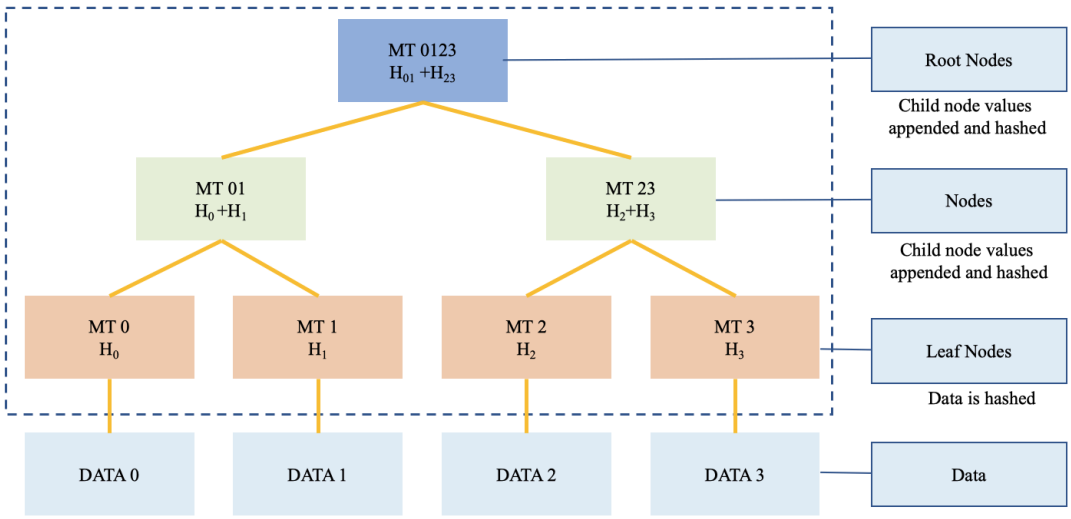

Figure 2.13 The Structure of Merkle Tree

Additionally, a binary tree structure is the most typical MT structure; any part in the binary tree structure has the relationship shown in Fig 2.14 (Hu & Gharavi, 2014). In Fig 2.14, *A* node is the parent, *B* and *C* are the children of *A* node; if taken *B* node as reference, *C* is the sibling of *B*. *B* and *C* are also interchangeable; that is, if taken *C* node as reference, then *B* node is the sibling of *C*.

If kept going to the next level, it will follow a similar relationship. Among the three nodes, *B* node is the parent, *D* and *E* nodes are the children; if taken *D* node as the reference, *E* node is the sibling of *D*, *D* and *E* nodes are interchangeable, too.

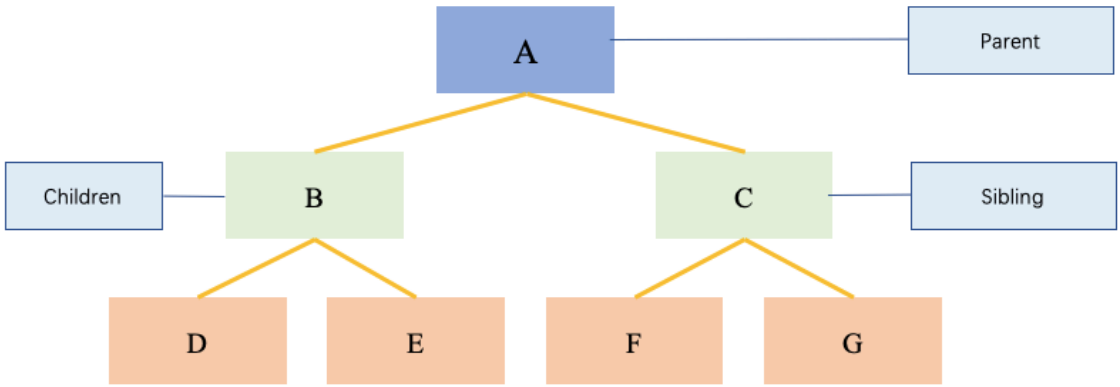

Figure 2.14 The Relationship Between Each Node in Merkle Tree

2.4.2 Hash

Hash is a function reference, as explained in Fig2.3, which maps arbitrary length data to a fixed-length data. For data integrity verification, the easiest way is to hash the entire data, obtain a fixed-length hash value, and publish the obtained hash value on the network so that after the user downloads the data, the data is hashed again. The result is compared with the hash value published on the Internet. If the two hash values are equal, the downloaded data is not damaged.

It can be carried out because a slight change in the input data may cause the hash to be unrecognizable, it is difficult to reverse the characteristics of the original input data based on the hash value (Bertoni, Daemen, Peeters, & Van Assche, 2014). If we are downloading it from a stable server, it is advisable to utilize a single hash. However, if the data source is unstable, once the data is corrupted, it needs to be re-downloaded. The efficiency of this download is very low.

2.4.3 Hash List

When data is transmitted in a peer-to-peer network, it is downloaded from multiple machines at the same time, which is considered unstable or untrustworthy. In the case of verifying the integrity of data, a better way is to divide the large file into small data blocks (e.g., when the original data to be transmitted is 1Gb or more significant, the file can be evenly divided into 1024 copies, each size of the serving is 1Mb). The advantage of this is that if the small block of data is corrupted during the transfer, we do not have to re-download the entire file.

To determine whether each piece of the segmented data is corrupted during transmission, it is only necessary to hash each block of data. When downloading BT, we need a list of hashes. In order to ensure the integrity and undamaged of the hash list, it is

necessary to put the hash values of each small block together and perform another hash operation on the long string so that the root hash of the hash list is obtained (Top Hash or Root Hash) (Munoz, Forne, Esparza, & Soriano, 2004). As indicated in Fig 2.15, when downloading data, we first get the correct root hash from the trusted data source, we can apply it to verify the hash list, then check the data block through the verified hash list.

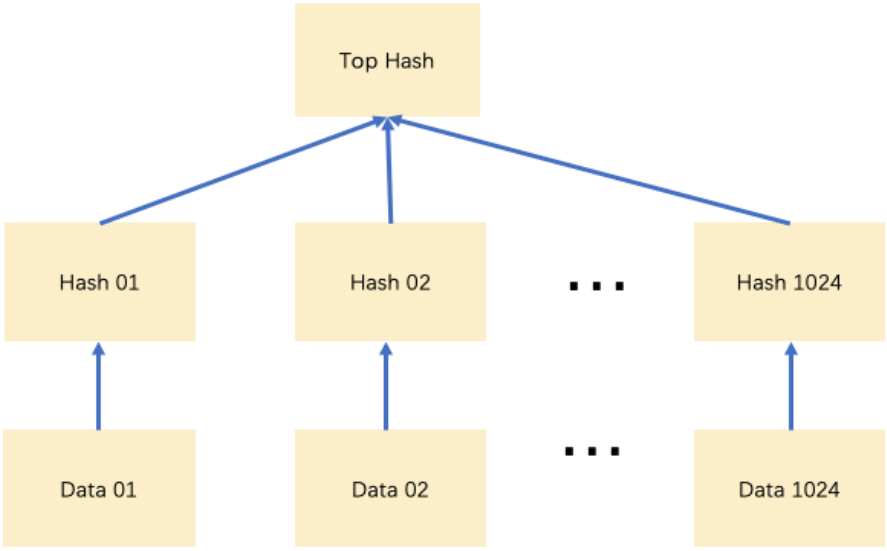

Figure 2.15 Hash List

2.4.4 Merkle Tree

Merkle tree can be seen as a generalization of a hash list, and the list can be seen as a particular Merkle tree, e.g., a binary Merkle tree has a height of 2.

Like the hash list, we split the data into small chunks of data with corresponding hashes and similar ones. Instead of directly computing the root hash, we merge the two adjacent hashes into a string, and then operate the hash of the string, so that every two hashes merge and have children. If the number of hashes in the bottom layer is singular, then a single hash must appear in the end. In this case, it is hashed directly so that we can get its child hash. For pushing up while still using the same way, we can get a new number of the new hashes and eventually form an upside-down tree. At the root, this generation has a root hash. We call it Merkle root (Liu, Ranjan, Yang, Zhang, Wang, & Chen, 2015).

Before downloading the network from the P2P network, we obtain the Merkle root of the file from a trusted source. Once we have the root of this tree, we can get the tree from other sources that are not trusted. The received Merkle tree with a trusted root will be checked. If the Merkle tree is corrupt or fake, we get another Merkle tree from another source until we get a Merkle tree that matches the root of the tree (Suat ÖZDEMİR, 2009).

The main difference between Merkle tree and the hash list is that we can download and immediately verify a branch of the Merkle tree (Mao, Zhang, Li, Li, Wu, & Liu, 2017). Because the file can be divided into small data blocks, if there is a piece of data corruption, we only re-download the data block. If the file is enormous, it is difficult for the Merkle tree and the hash list to verify all the data at once, but the Merkle tree can be downloaded one branch at a time, and then immediately verify the branch (Andreeva, Bouillaguet, Dunkelman, Fouque, Hoch, Kelsey, Shamir, & Zimmer, 2016). If the branch is verified, the data can be checked. The hash list can only be verified by downloading the entire hash list.

2.4.5 Retrieve data with Merkle Tree

In order to understand and explain this problem more conveniently and intuitively, in Fig 2.16 and Fig.2.17, we assume that the target object is equally divided into 16 groups of data, the eighth group of data is damaged. Now, we need to find out which group of data has a problem in the shortest time and how to re-download the data. To ensure whether the complete transmission of data using the traditional method, a hash operation is conducted on each set of data and then hash the hash value generated by using each set of data again to generate a root hash. Because the verifier does not know which group of data has problem, the search is to start the traversal search in the order from M1 to M16, or from M16 to M1 in the inverse order; but no matter it is the former or the latter, its efficiency is extremely low.

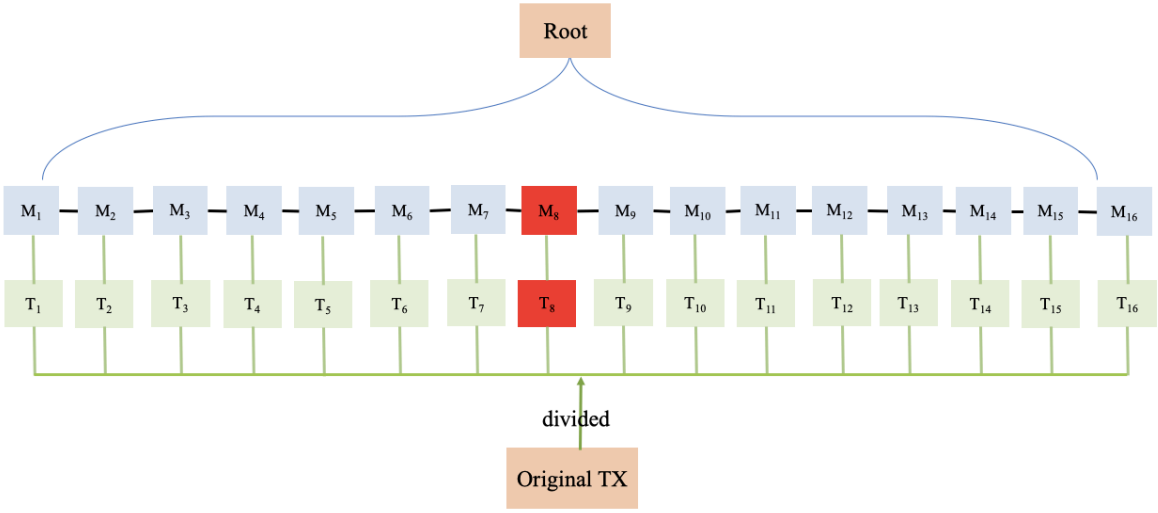

Figure 2.16 Classical Method

Next, let us look at whether the solution of this problem has improved after using the Merkle tree approach. As we see from Fig 2.17, the same 16 sets of data are generated and then hashed separately. Then, M1 and M2 are hashed again to generate a new parent node M1+M2; M3 and M4 are hashed to generate a new parent node M3+M4. By using an analogy, eight new nodes will be generated, and then two hashes will be generated to generate a new batch of nodes until the root node is finally generated.

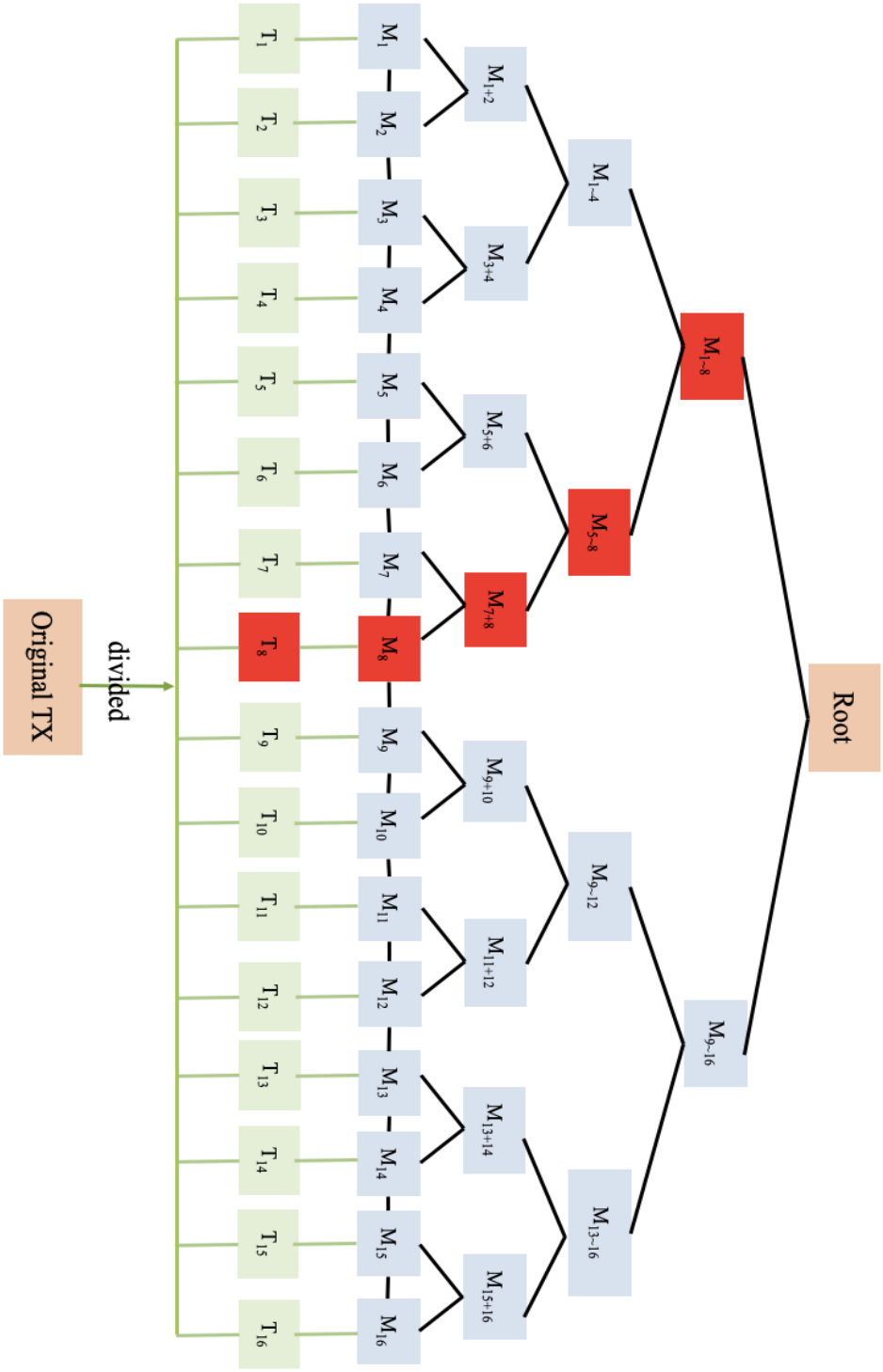

Figure 2.17 The Data Structure of Merkle Tree

When we need to confirm whether the transmission is wrong, the steps are shown in Fig 2.18. The first step is to compare the root notes. If inconsistent, then its children nodes M1~M8 and M9~M16 will be retrieved. In the second step, if M9 ~ M16 are the same, and M1~M8 are different, then the children nodes M1~M4 and M5~M8 of M1~M8 are compared. In the third step, if M1~M4 are the same, and M5~M8 are different, then the

children nodes M5+M6 and M7+M8 of M5~M8 are compared. In the fourth step, if M5+M6 is the same and M7+8 is different, then we compare the children nodes M7 and M8 of M7+8. in the fifth step, if M7 is consistent, M8 is different, and M8 is a leaf node, then its directory is obtained. The sixth step is to complete the search. Therefore, we can quickly get the problematic node.

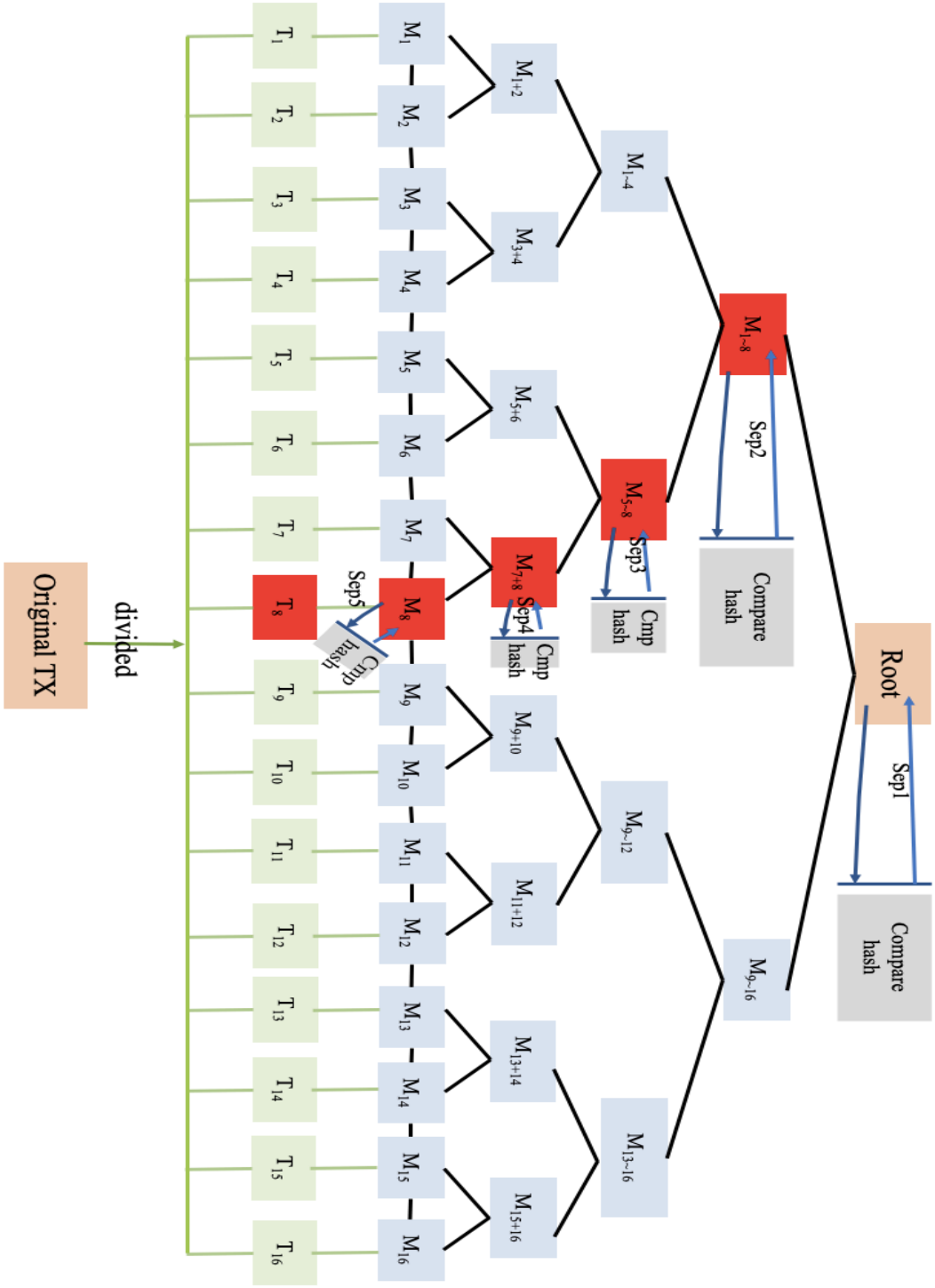

Figure2.18 Steps to Retrieve Data

2.4.6 Perfect Binary Merkle Tree

A perfect binary tree with the same structure of Merkle tree has the following properties:

- The number of leaves is always 2^n (n = 0,1,2,3…).
- Each node has 0 or 2 children.
- All leaves are on the same level.
- In a perfect binary tree, the following equations can be applied:
  - Total number of leaves: $L = (N + 1) / 2$.
  - Total number of nodes: $N = 2L - 1$.
  - Total number of levels: $L_V = \log_2(L) + 1$ and $L_V = (\ln(L) / \ln(2)) + 1$

In Fig 2.19, if a Merkle tree has only one leaf, this leaf is also the root.

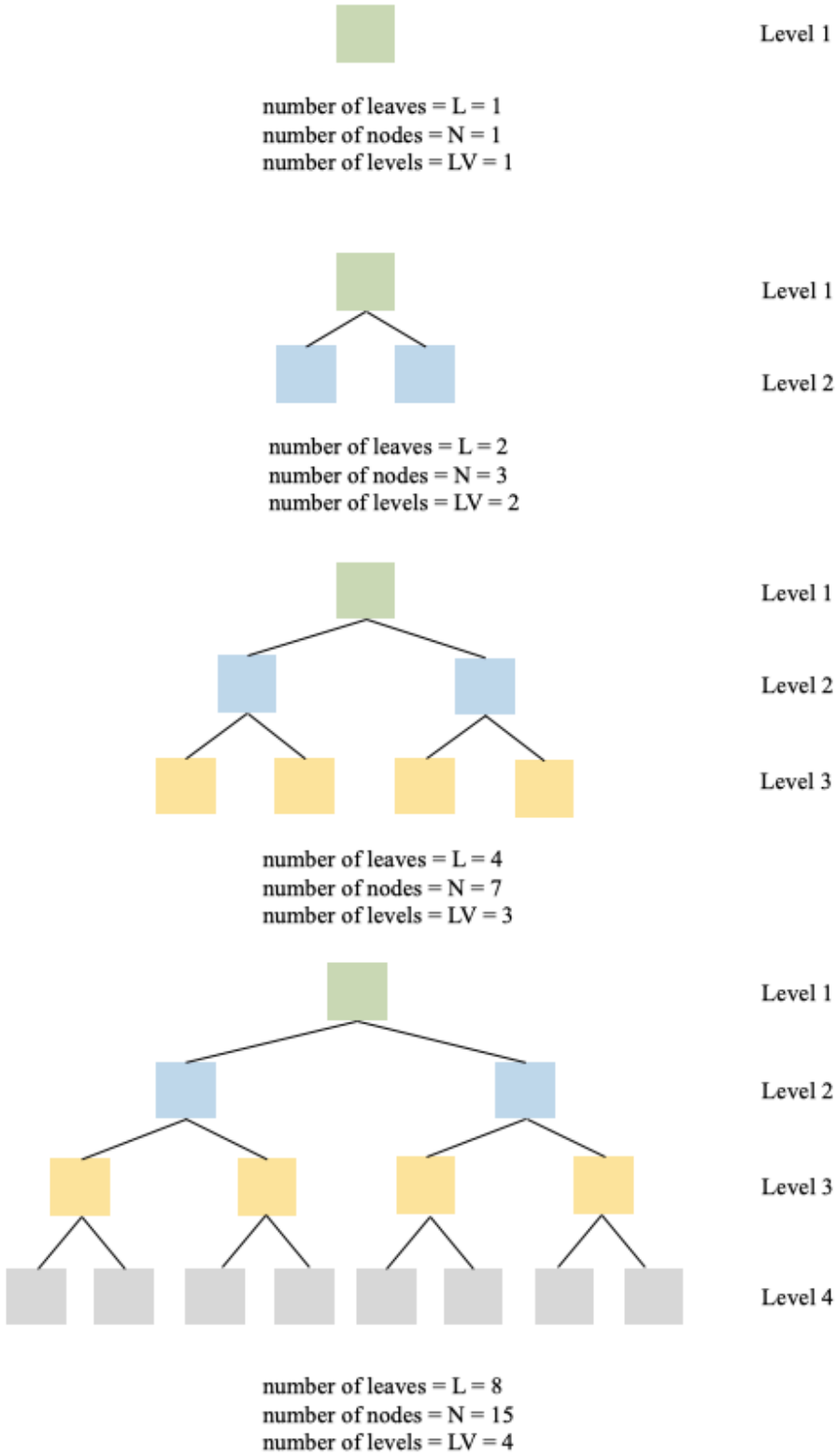

Figure 2.19 Perfect Binary Tree

Chapter 3
Methodology

This chapter mainly introduces details of research methods used in this thesis, how to achieve the research purpose based on this method. In this chapter, we will introduce the implementation of the private chain that satisfies our experiments and the application of the Merkle tree for big data storage. At the end of this chapter, how to store big data of videos in blockchain by using smart contracts will be addressed.

The preservation of big data on the chain always has a big problem in blockchain (Zhou, Wang, & Sun, 2018). Due to increasing enthusiasm of the major sectors in this field, the value of cryptocurrency keeps increasing all the time. In 2018, the exchange of major digital currencies into US dollars was shown in Table 3.1. Despite the recent decline in digital currency, the overall performance of ETH is still rising sharply throughout the life cycle. In Fig 3.1, it shows the Ether historical exchange with USD (https://www.blockchain.com). Fig 3.2 illustrates the tendency of ETH development recently.

Table 3.1 Cryptocurrency exchange with USD

Coin	Weight	Last Price(USDT)
BTC	35.89%	3992.18
ETH	16.22%	110.37
BCH	5.24%	151.32
LTC	2.68%	31.40
ETC	2.24%	4.54
EOS	16.24%	2.44

Figure 3.1 Ether Historical Exchange with USD

In this thesis, only real situations are simulated. As the value of digital currency continues increasing, the number of participants has experientially grown, which triggered three issues.

First, Satoshi has issued the electronic currency from the beginning. The hash collision is justified in real time to guarantee that a new block is generated every 10 minutes (Underwood, 2016).

Second, a large number of blockchains started up and followed the Bitcoin to issue a new cryptocurrency. To protect the rights of investors, the Ethereum development team has fulfilled the previous Ethereum blockchain (Mehar, Shier, Giambattista, Gong, Fletcher, Sanayhie, Kim, & Laskowski, 2019).

Third, in recent years, due to the increase of cryptocurrency, more and more people are investing in bitcoin mining. Besides, due to the increase of participants in mining, the number of blocks has increased, the volume of transaction data that needs to be stored in the leading network of the blockchain is also soaring out of our imagination. Blockchain of several mainstream cryptocurrencies in Table 3.1 will take a huge time to finish data synchronizing (Kewell & Michael Ward, 2017).

In summary, this thesis uses the method of constructing a private chain to test our experiments. Also, in order to inherit the commonality of many mainstream blockchain, the private chain also accompanies with high efficiency and low investment due to its flexibility and freedom.

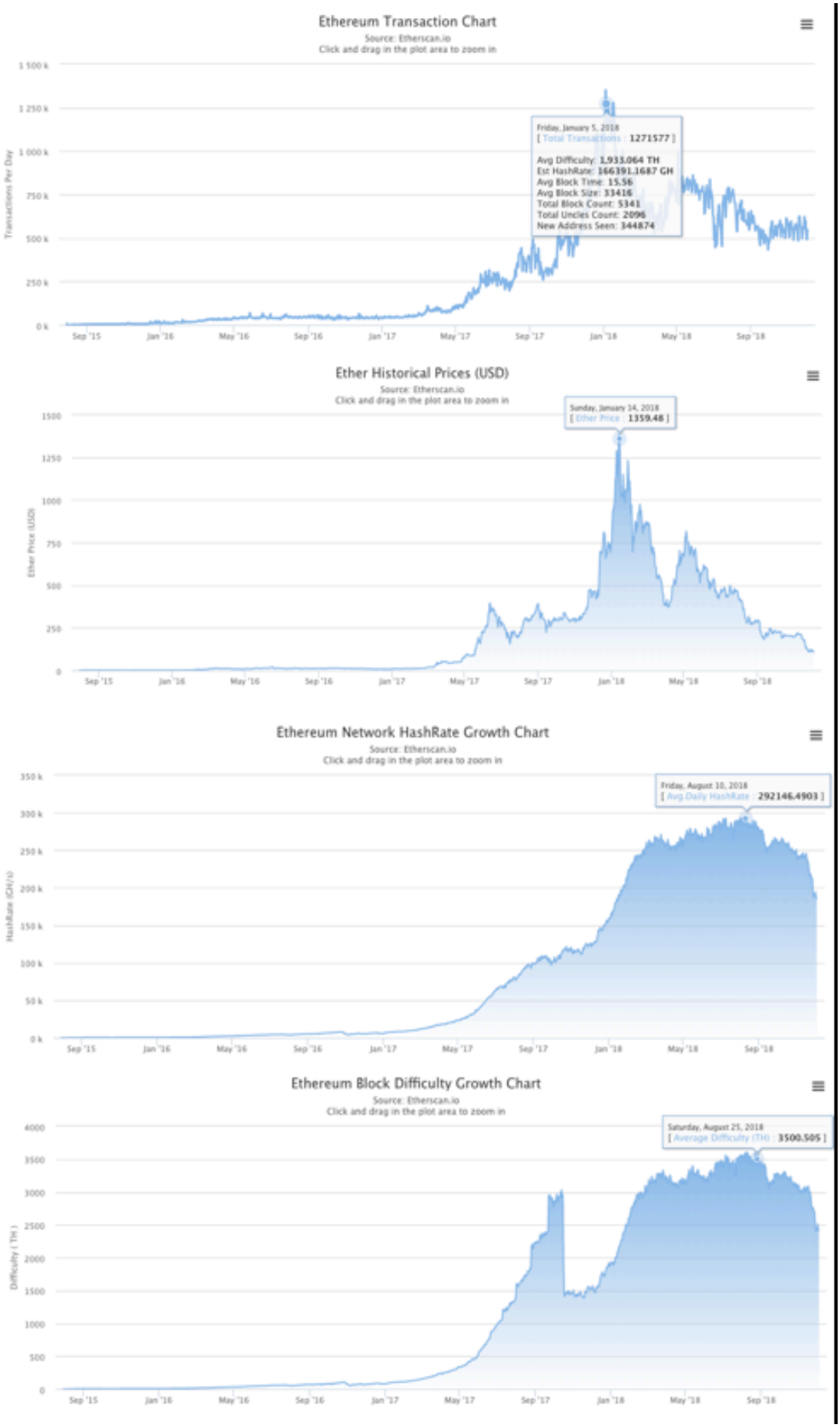

Figure 3.2 Statistical Data

3.1 Design

For the sake of saving costs and improving efficiency, this project will adopt the method of creating a private chain for testing. In the following sections, how to construct a private chain will be described in detail. In this chapter, the ideas, experimental environment, and experimental steps will be detailed in after all.

3.1.1 The design of this project

As shown in Fig 3.3, we see that usage rate in Bitcoin blockchain has been kept at a high level since 2015, which indicates the urgent need for block storage.

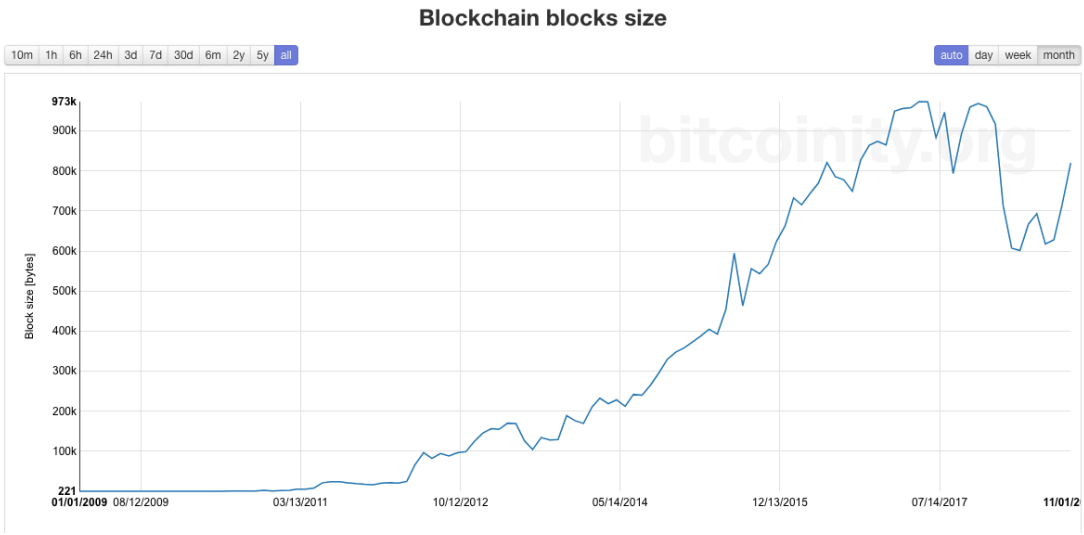

Figure 3.3 Block Size Changes

Although the demand for block expansion is vast, there are many organizations in the world conducting relevant research, but there is not much progress, because everyone considers that expanding the capacity of the block itself will delay the transaction rate. There is a delay in the entire blockchain (Dai, Zhang, Wang, & Jin, 2018).

In this thesis, we fully take into consideration the data written into the blockchain while considering the transaction of the blockchain. We need to introduce the idea of a Merkle tree; as long as the data exceeds the standard, it will be cut to a standard size so that we can guarantee that everyone does not exceed the maxima of standard. Next, every two adjacent objects will be hashed until a unique hash value generated, which is called

root hash value. Finally, the root hash value is written into the block to achieve permanent storage on the chain. As shown in Fig 3.4, the target data is split into eight equal parts, and the Merkle tree is used to generate a root for the two hashes, which is saved in the header of the block. However, this does not fundamentally solve the problem of data saving. Only the hash value will be stored.

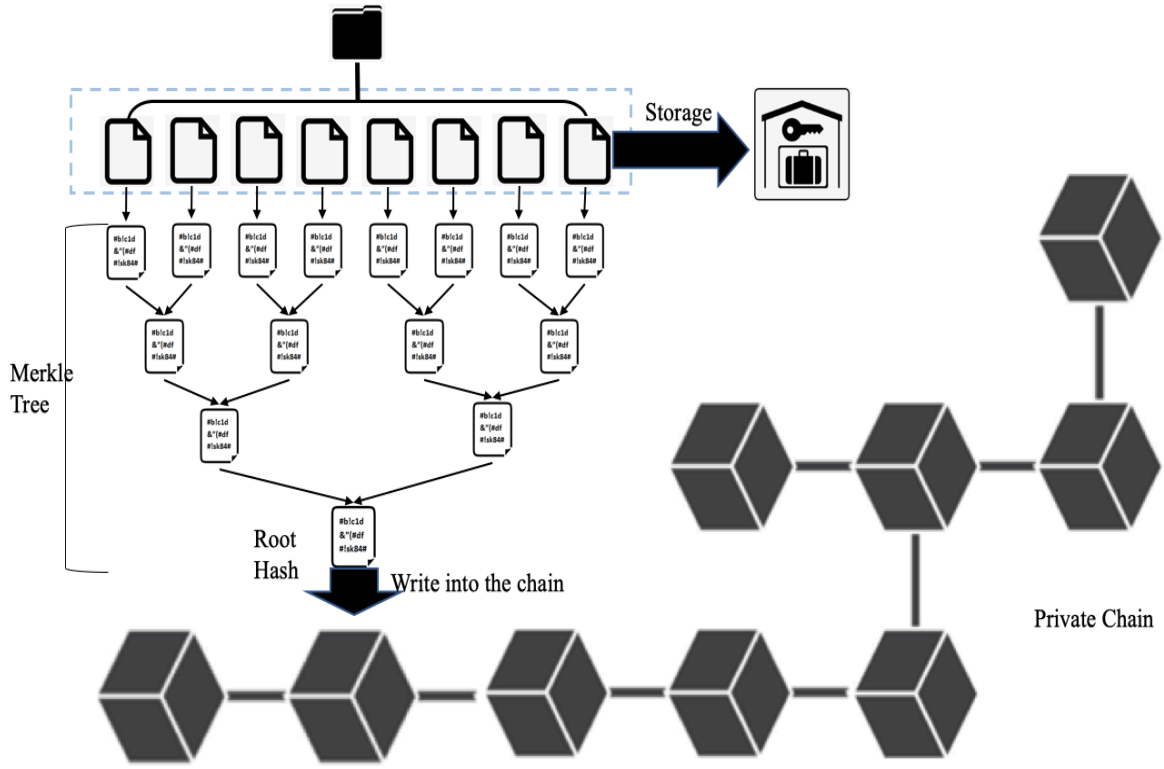

Figure 3.4 Preliminary Design

Regarding the preliminary design, the target data is only stored in the local or online database or a permanently saved address, the real chain is saved as a hash value. This method theoretically achieves permanent preservation of the chain but has a plenty of drawbacks and limitations.

Since the hash is characterized by a one-way output of a fixed-length string, the reverse cannot be traced; therefore, the original data cannot be directly read through the hash value. When the operation is completed, the root hash value is permanently stored in the blockchain and has a one-to-one correspondence with the original data. When we need to read or operate the data saved at that time in future, the first step is to confirm whether the data is the original one. When a file is stolen, or a significant event needs to

be extracted from the video at the time, the hash value of our original video has been stored on the blockchain.

When we retrieve the complete video data from the places where the video data was stored initially, the purpose of ensuring the data has not been tampered or lost. We can perform a hash operation on the data and then compare the two hash values. If they are consistent, the data is original one which is not possible to be damaged or tampered. Otherwise, we need to return to the place where the data was initially stored and then find the relevant data. The process is shown in Fig. 3.5.

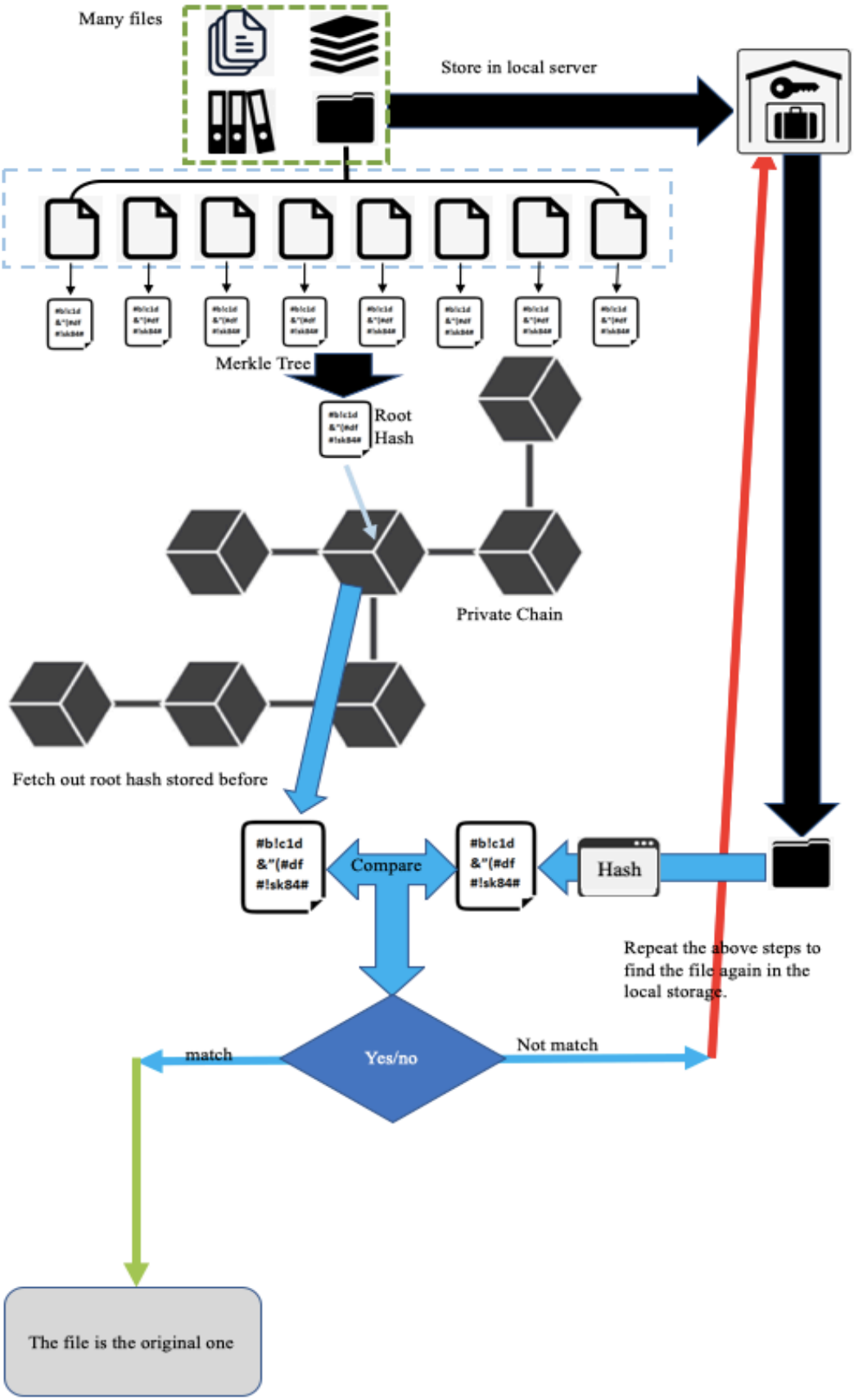

Figure 3.5 The Function in This Design Method

However, it is easy to find that there are several problems as follows. One is to store the root of big data instead of the data itself, the second is that data storage is easy; but in the latter, only one comparison and confirmation function can be achieved in the later stage, it cannot be saved and retrieved in real time.

Combined with the existing shortcomings and existing loopholes in preliminary design, the improved design scheme is indicated in Fig 3.6. There are also a lot of original files that need to be stored in the chain. Therefore, one of the files is split into eight parts and hashed separately to generate the leaf nodes of Merkle tree. According to the data structure, a unique root is finally generated. The root hash is produced correspondingly and saved on the chain. Once the root hash is corrupted, its data will be permanently saved and cannot be tampered with. This ensures uniqueness of the original data.

At the same time, unlike the idea of preliminary design, we no longer store the original data locally, but instead put the eight pieces of data after it has been split into the P2P network. Each node can be chosen to save one or more pieces of data, we assume that each node can only be saved individually.

When a customer needs to retrieve the document after a period of time, the following operations are performed. The client queries the private chain for the related root hash stored at that time; the root hash branches downward to finally obtain the hash value of the leaf node corresponding to the original file and then utilizes its addressing attribute to index in the P2P network. Because the hash value has a one-to-one correspondence with the file, the split file of the original file is found; finally, the complete file can be composed.

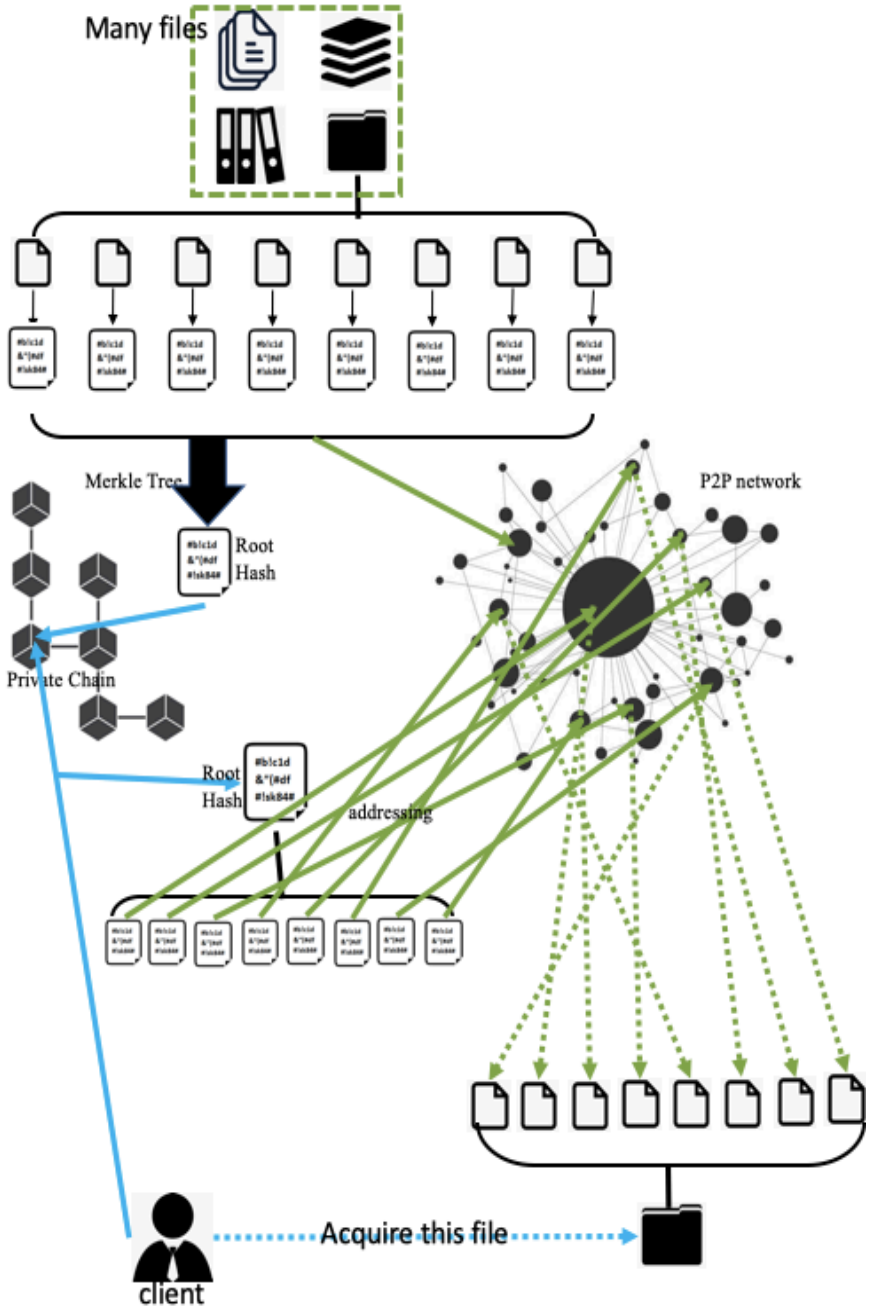

Figure 3.6 Further Design Method

3.2 Private Chain Implement

In this thesis, considering the incompleteness of the underlying technology of Bitcoin, we will optimize the current relatively wide-ranging and Turing-complete Ethereum to build a private chain for installation and debugging (Gramoli & Staples, 2018).

3.2.1 Building Software-Testing Environment

The Ethereum client is used to access the Ethereum network for account management, trading, mining, and smart contract related operations. Currently, there are multi-language client-based implementations. Go-ethereum client, i.e., Geth, which was implemented in Go language, supports access to the Ethereum network and becomes a full node (Pãnescu & Manta, 2018). It also provides a JSON-RPC interface as an HTTP-RPC server.

Ethereum supports programming languages for two smart contracts: Solidity and Serpent. The Serpent language faces security issues and is now deprecated. The Solidity syntax is similar to JavaScript and the compiler Solc can compile the smart contract source into a binary code that the Ethereum virtual machine EVM can be executed. Now, Ethereum offers a much convenient online IDE-Remix.

Remix eliminates the need to install Solc and its build process, which provides the binary and ABI required to deploy the contract directly. This method will also be utilized in this thesis.

3.2.2 Configuring the Initial State

Before running the Ethereum private chain, we first need to define the first block of the chain, which is the genesis block. From now on, the relevant information of the creation zone is written into a JSON format configuration file, described as follows.

```
{
    "config":{
    "chainID":1234,
    "homesteadBlock":0,
    "eip155Blcok":0,
    "eip158Block":0
    }
    "alloc": {},
   "coinbase":"0x0000000000000000000000000000000000000000",
```

```
        "difficulty": "0x4000",
        "extraData": "0x0",
        "gasLimit":"0x2fefd8",
         "nonce":"0x0000000000000042",
        "mixhash":
          "0x0000000000000000000000000000000000000000000000000000000000000000
    0000",
       "parentHash":"0x0000000000000000000000000000000000000000000000000000000
0000000000000",
        "timestamp": "0x00"
      }
```

The concepts in this structure are explained (Maxim, Anastasiya, Sergey, Yuri, & Anastasiia, 2018):

- **ChainID** specifies a separate blockchain network ID. The function of this network ID is connecting to other nodes. The network ID of the Ethereum public network is 1. In order not to conflict with the public chain network, we must specify our own network ID when running the private chain node. Nodes of different ID networks cannot be connected to each other.
- **Alloc** is applied to pre-set the number of accounts and the amount of Ethereum in the account. Because the private chain mining is relatively easy, we do not need to pre-set the account with the coin, and create it when needed.
- **Coinbase** is the miner's account number,
- **Difficulty** value of mining
- **ExtraData** refers to additional information, here we can fill in personalized data
- **gasLimit** sets the total consumption limit for GAS and can limit the sum of transaction information that the block can contain.
- **Nonce** is a 64-bit random number utilized for mining. Note that the setting and MixHash need to meet the conditions described in Ethereum's Yellow paper.

- **Mixhash** is used in conjunction with nonce for mining, which is a hash, generated by a portion of the previous block. Note that this and nonce settings need to meet the conditions described in Ethereum's Yellow paper.
- **ParentHash** is the hash value of the previous block, as it is a Genesis block, so this value is 0.
- **TimeStamp** is also 0 since it is a Genesis block.

3.3 Elliptic Curve Digital Signature Algorithm

3.3.1 Overview

Encryption algorithms are widely applied to digital cryptocurrency systems. Amongst them, elliptic curve digital signature algorithm (ECDSA), which is widely optimized and recognized as the highest safety factor, is based on evolution of digital signature algorithm (DSA) and elliptic-curve cryptography (ECC) (Wang, He, & Ji, 2017). The simplified equation shown in Eq (3.1), is also known as the Weierstrass normal form.

$$Y^2 = X^3 + aX + b \tag{3.1}$$

Given a = -5, b = 5, it is shown in Fig 3.7.

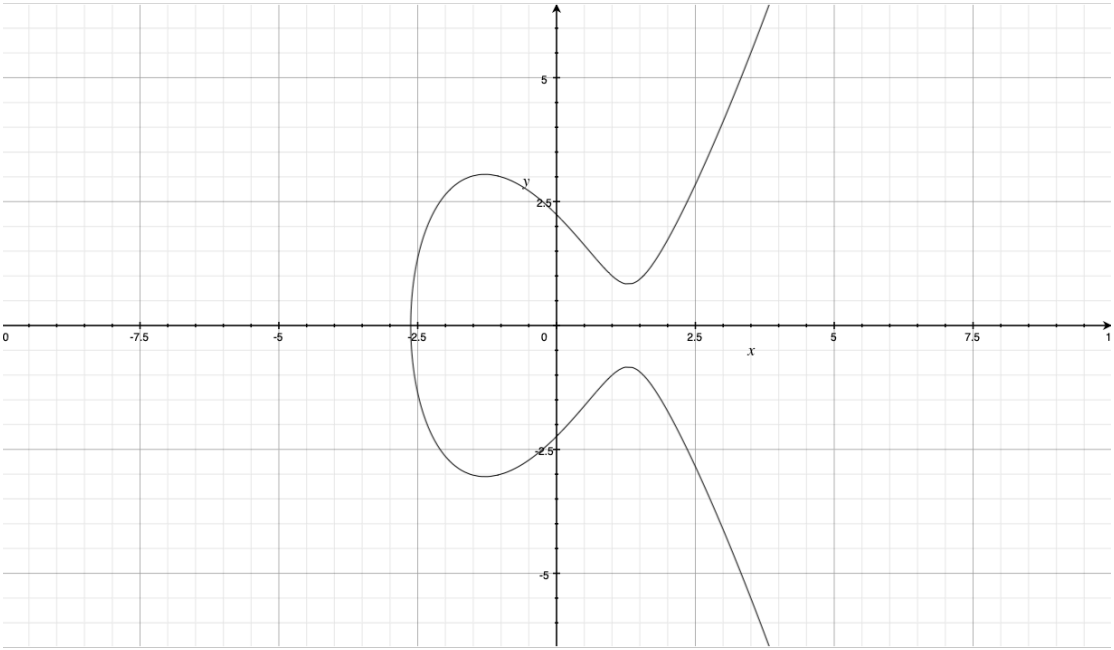

Figure 3.7 Elliptic Curve $Y^2 = X^3 - 5X + 5$

Given a = -7, b = 3, it is illustrated in Fig 3.8.

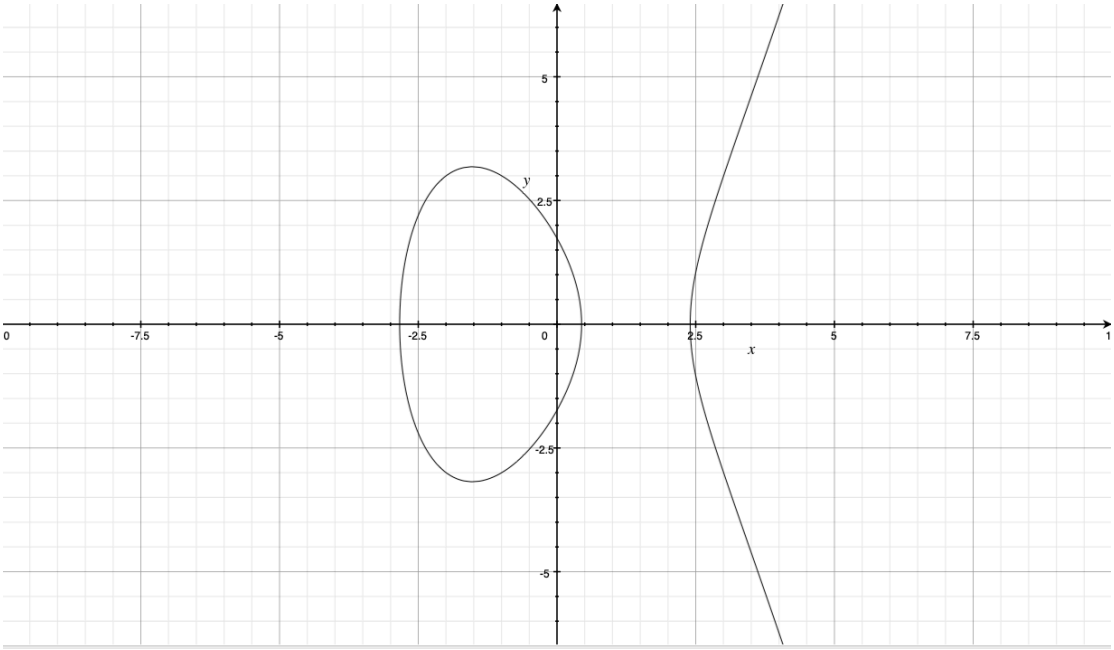

Figure 3.8 Elliptic Curve $Y^2 = X^3 - 7X + 3$

In Fig 3.7 and Fig 3.8, a and b directly affect the whole curve. At this time, we assume that there are two points **A** and **B** on the curve whose coordinates are **A** (Xa, Ya) and **B** (Xb, Yb) shown as Fig 3.9, a straight line passing through points **A** and **B** intersects the curve at point **C** (Xc, Yc). Therefore, these three points should have the following relationship **A** + **B** + **C** = 0. Regarding the symmetric point **C**' of point **C** concerning the x-axis, there should have **A** + **B** = **C**'.

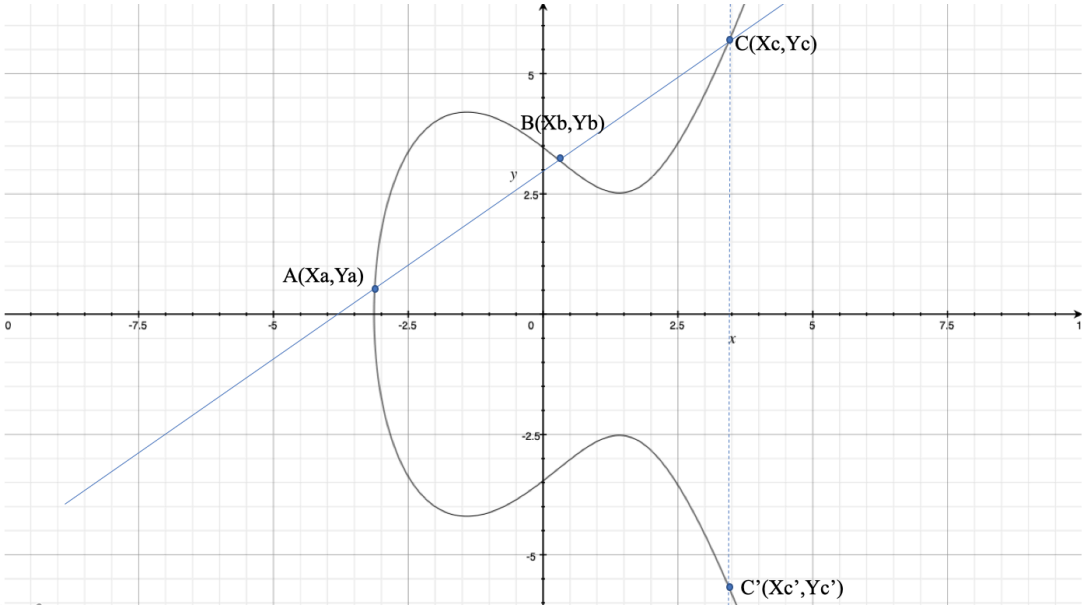

Figure 3.9 Elliptic Curve Intersect a Line with A and B

However, there is a particular case, that is, when the two points **A** and B coincide, as illustrated in Fig 3.10, there is 2**A** = **C**'.

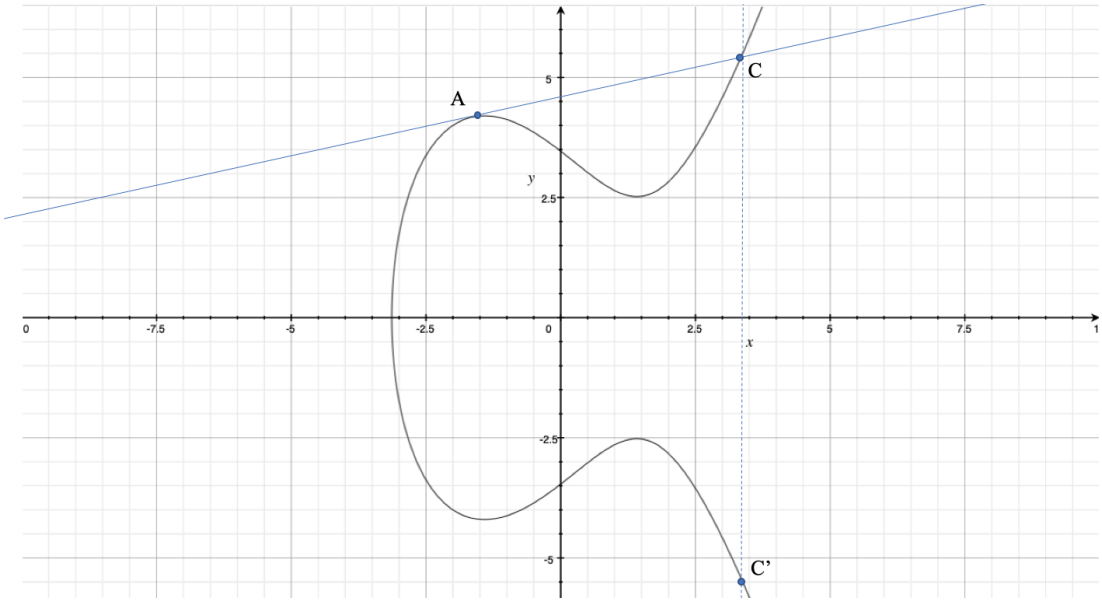

Figure3.10 Elliptic curve intersect one line when **A** = **B**

Further, from point **C**', a tangent line intersection point is taken from the curve to take **D**', which takes a symmetry point **D** with respect to the *x*-axis. Similarly, **D** = 2**C**' = 4**A** can be obtained. That is, we take a little **G** on the curve and make a tangent line to the curve through **G**. The obtained intersection point is symmetric concerning the *x*-axis, then 2**G** can be obtained, the operation can be repeated, given 4**G** and 8**G** as shown in Fig 3.11.

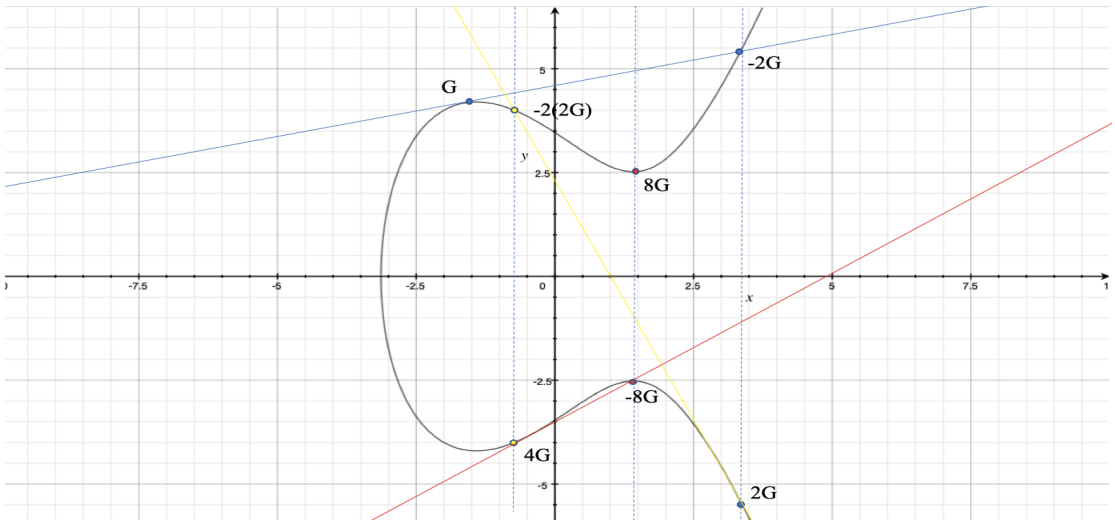

Figure 3.11 Elliptic curve with 8**G**

We see that when **G** point is given, and the multiple *n* is known, it is not difficult to find *n***G**; if *n***G** is known, in turn, this will be very difficult to get the value of *n*.

3.3.2 Applications of Elliptic Curve Algorithm on Blockchain

An elliptic curve is relatively intuitive to reflect its mathematical principle. In the digital currency encryption algorithm, because integers are usually adapted, we usually use Eq (3.2) as follows (Stewart, Ilie, Zamyatin, Werner, Torshizi, & Knottenbelt, 2018):

$$Y^2 = \ (X^3 + aX + b) \ \text{mod p} \qquad (3.2)$$

where $(0, p)$ is the interval of Y^2.

Compared to DSA based on RSA cryptography, the length of public key required in digital signatures can be greatly reduced. For example, we propose a digital signature with a security level of 80 bits, the public key length based on ECDSA is almost twice of the security level, the required public key length of the RSA under the same security level is at least 1024. The length of the signature generated by the algorithm based on either ECDSA or RSA is about 320 bits; thus, the advantage of ECDSA related to RSA is distinct.

3.3.3 Implementation

Using elliptic curve cryptography is complicated to know that **G** and *n***G** should reverse *n*, *n* is the private key, and *n***G** is the public key (Suárez-Albela, Fraga-Lamas, & Fernández-Caramés, 2018).

(1) Set the public key **PK** and the private key *pk*.

(2) The process of public key encryption is to select a random number *N* and the data *D*, in order to generate ciphertext *C*, **PK** is a coordinate point of the public key;

(3) The process of decrypting the private key is, $D + N_{PK} - pk(\mathbf{NG}) = D + N(pkG) - pk(\mathbf{NG}) = D$

(4) The process of creating this private key is to select the random number N and calculate the point **NG** (x, y), $S = (h + pkx)/N$ according to the random number N, the data hash, and the private key. The data and the signature {**NG**, S} are sent to the recipient.

(5) The process of public key verification is that the receiver receives the data and the signature {**NG**, S}, obtains the hash value of the Data; now we need to calculate the $hG/S + x\mathbf{PK}/S$ using the sender's public key **PK** with the **NG**. For comparisons, if the results are equal, the signature verification is passed. It satisfies:

$$hG/S + xPK/S = hG/S + x(pkG)/S = (h + xpk)\ G/S = N\ (h + xpk)\ G\ /\ (h + pkx) = NG \quad (3.3)$$

3.4 P2P network based on IPFS technology

3.4.1 Concepts

Interplanetary File System (IPFS) is a global, peer-to-peer distributed version of the file system, the goal is to supplement (or even replace) the hypertext transfer protocol (HTTP) that currently dominates the Internet, all with the same file system (Chen, Li, Li, & Zhang, 2017). The computing devices are connected. The principle replaces the domain-based address with a content-based address; that is, the content that the user is looking for is not a real address but is stored in a fixed place, which does not need to verify the identity of the sender, but only requires to verify the hash of the content. It can make web pages faster, safer, and robuster (Alessi, Camillo, Giangreco, Matera, Pino, & Storelli, 2018).

3.4.2 Functions

In Fig 3.6, we explained that we want to build a P2P decentralized database for storing data, which can be done based in the IPFS system. A computer with IPFS installed is equivalent to a node based on the whole network. The local computer has certain hard disk storage, which can share local video or video data in the network. It can also download the required resources from other nodes in the network (Zheng, Li, Chen, & Dong, 2018).

After installing IPFS locally, we can display the ID of the local machine and the information about the Public Key through the IPFS id as described in Fig 3.12.

Figure 3.12 Local ID

Also, as a node with the storage capacity in the network, we can freely adjust the size of the space stored as a network hard disk; as explained in Fig 3.13, we enter the command “export EDITOR=/usr/bin/vim” and “ipfs config edit”. Upon setting the interface, we see that the value of default storage is 10GB, where this parameter can be adjusted according to its status.

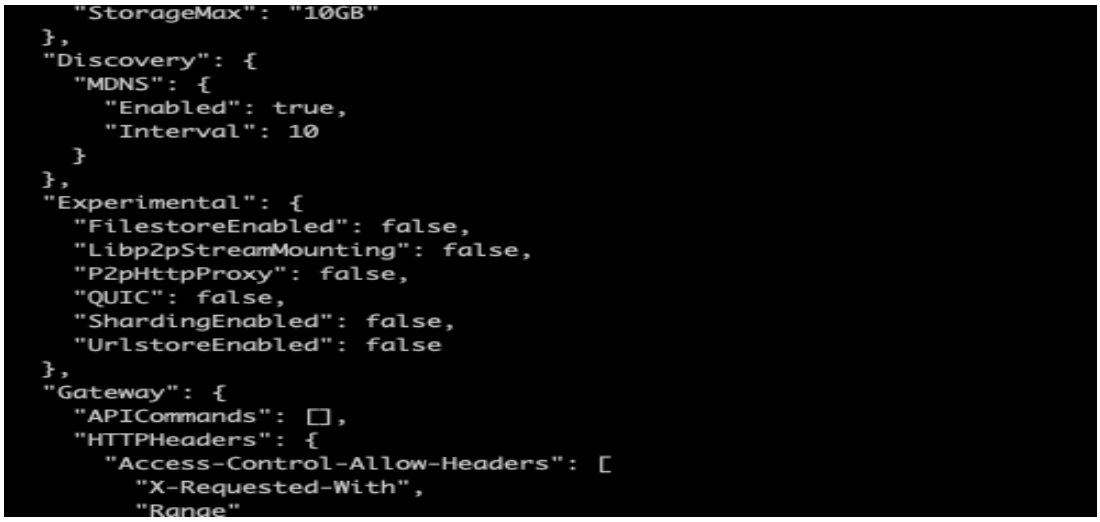

Figure 3.13 Set StorageMax

3.5 Merkle Tree Applied

3.5.1 Merkle Tree Applied in This Thesis

We have explained the principle of Merkle tree in this thesis, and we will apply this technology to dispose of video data. Like Fig 3.14, we parse the object video data, considering that the nature of videos is composed of one frame in the order of temporal

sequence, in order to ensure the efficiency of transmission and the quality of transmission, we split the video in equal duration. The original video data is equally divided according to its size, and we set the last one which cannot reach the standard size as the remaining part. Then the hash values are computed from each video data, and we save these hash values and their corresponding video data into the distributed database IPFS.

Next, we hash the hash values obtained from the previous round and hash the hash values obtained in this round again. Repeated these steps, finally, we will get two hash values, the root hash value of this video file will be generated after this round. We save the root hash to the blockchain.

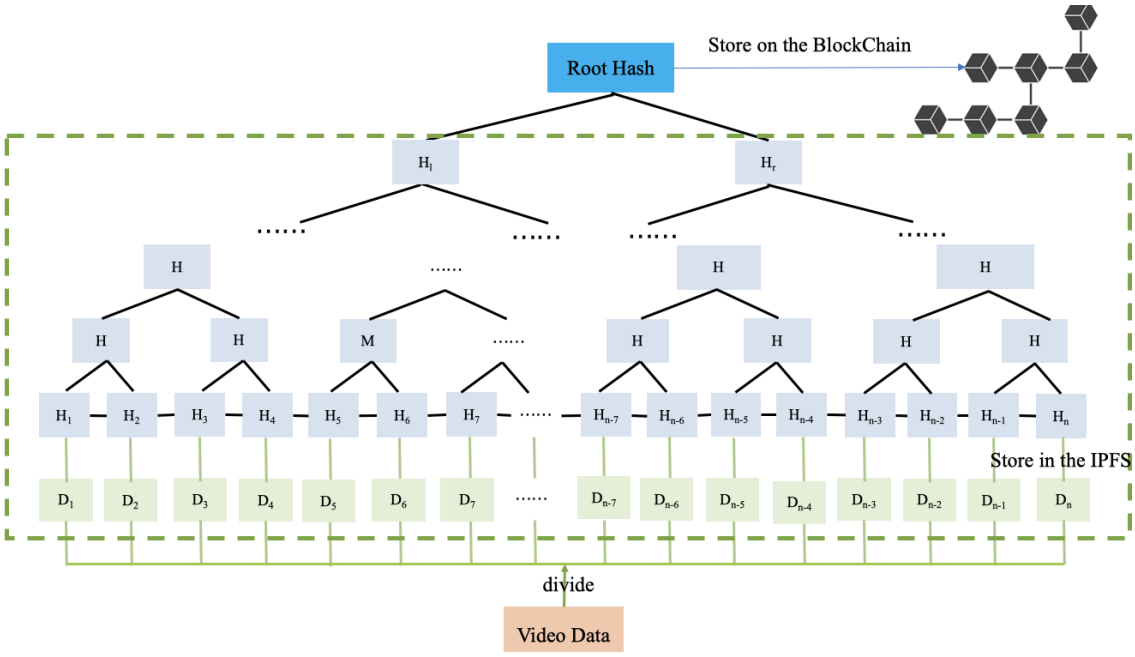

Figure 3.14 Merkle Tree for Video Data

3.5.2 Implement Method

The principle diagram of using a Merkle tree is depicted in this thesis. Next, we will discuss it on details. We will introduce the core parts of this piece of source code.

```
{
public MerkleTrees(List<String> txList) {
    this.txList = txList;
    root = "";
}
```

```
public void merkle_tree() {
        List<String> tempTxList = new ArrayList<String>();
             for (int i = 0; i < this.txList.size(); i++) {
           tempTxList.add(this.txList.get(i));
        }
        List<String> newTxList = getNewTxList(tempTxList);
        while (newTxList.size() != 1) {
           newTxList = getNewTxList(newTxList);
        }
this.root = newTxList.get(0);
    }
private List<String> getNewTxList(List<String> tempTxList) {
List<String> newTxList = new ArrayList<String>();
        int index = 0;
        while (index < tempTxList.size()) {
            String left = tempTxList.get(index);
           index++;
            String right = "";
           if (index != tempTxList.size()) {
              right = tempTxList.get(index);
           }
            String sha2HexValue = getSHA2HexValue(left + right);
           newTxList.add(sha2HexValue);
           index++;
        }
        return newTxList;
    }
}
```

We will dispose of all the object video data following four steps:

Step 1. Construct a function of Merkle tree.

Step 2. Execute the function and set a root value.

Step 3. List the last two hash values and named them with left and right.

Step 4. Get the root hash of the object.

3.6 Deploy the Smart Contract

So far, we have created our private chain and prepared a distributed online database IPFS for storing video data. We already designed the application of the elliptic curve encryption algorithm, and the target video data has been already disposed of, the root hash has been generated as well. The following step is to store the hash value into the blockchain.

Considering the hash value is just a character string, we need to store this hash, including the complete information of the videos.

Because deploying smart contracts requires an amount of eth, we must set up our e-wallet first. MetaMask, a kind of Electrum wallet, is widely used and easy to be installed. It can be directly utilized as a plugin in Google Chrome (Chen, Xu, Gao, Lu, & Shi, 2018).

In order to avoid unnecessary expenses and achieve the goal of smart contract deployment, we choose localhost:8535 to test the network and connect the wallet with the local private chain.

3.7 Blockchain Applied in the Online Video Website

We repeat the above steps to store the video files which need to be stored in a blockchain. This project takes “American Best Dance Crew Season 7” as an example, and it has ten episodes in this season. We stored the first to tenth episodes orderly on the blockchain and optimize the blockchain that holds the video data as the backend database. We make a front interface simulating a simple online video website. The relevant search result can be extracted from the backend database. The main interface of the online video website is illustrated in Fig 3.15.

Figure 3.15 Main Page of Our Design

Chapter 4

Results

The experimental results related to the experimening will be listed in this chapter, the video data is stored in a blockchain through smart contract deployment successfully, the online video website based on the sorted playlist will be shown. Additionally, the cost and delay of this experiment will also be demonstrated; moreover, limitations of this project will be stated at the end of this chapter.

4.1 Settle the Object Video Data

4.1.1 Settle the Object Data of Videos

The primary method to present video data in this project is to apply a Merkle tree, as indicated in Fig 4.1. The size of the videos is 66.70MB. We see that it finally generates the root hash: “QmX6vJiVGNd1VtwnLEr AuEoPfgtW9TzabqkjWmx8y1zpvr”. It is evident that two hash values have been linked to this root hash; we can check what contains in each value related to the root hash.

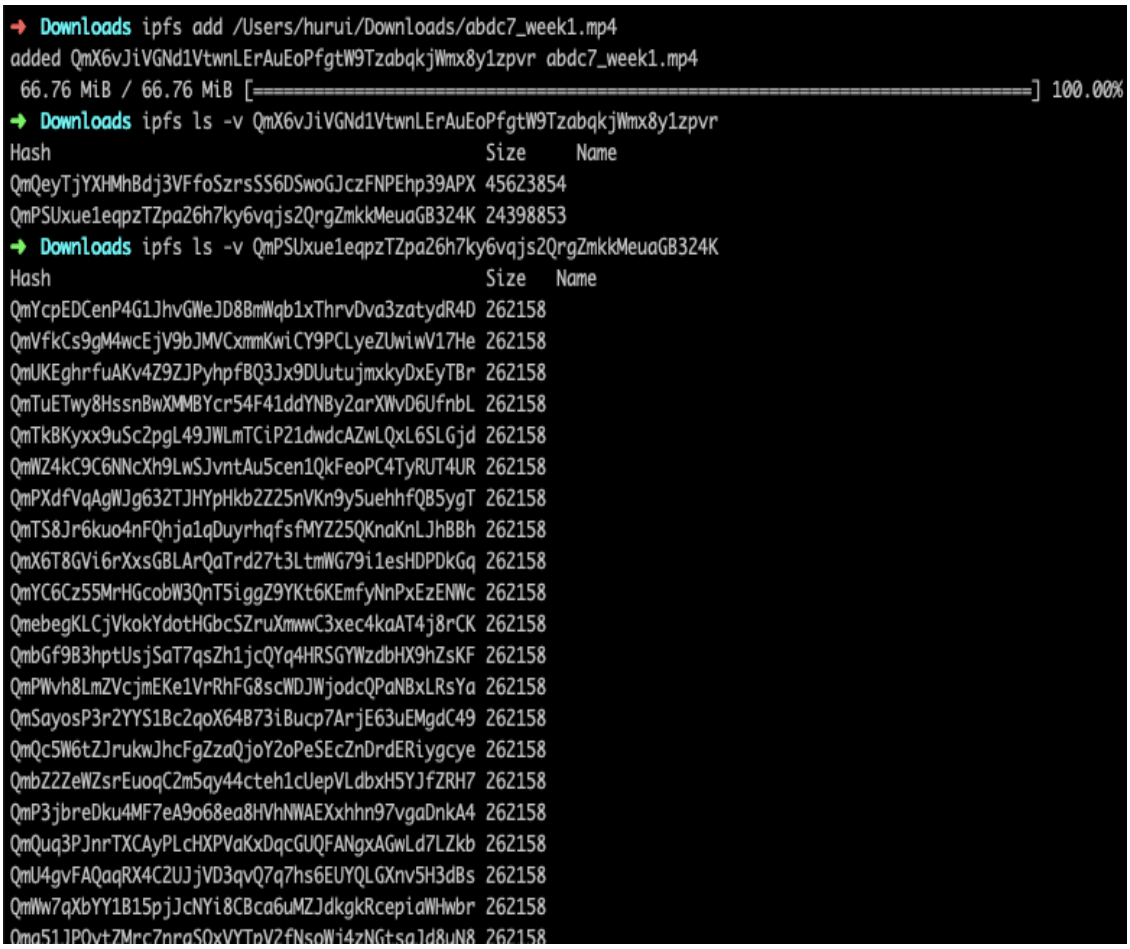

Figure 4.1 Divide the data of video

4.1.2 Deploy the Smart Contract

After the smart contract is compiled automatically, the system will call the e-wallet; meanwhile, a prompt window will be popped up to show how much ETH will be spent on the deployment of the smart contract.

4.1.3 Store the Object of Videos on the Blockchain

The interface of our project is shown in Fig 4.2.

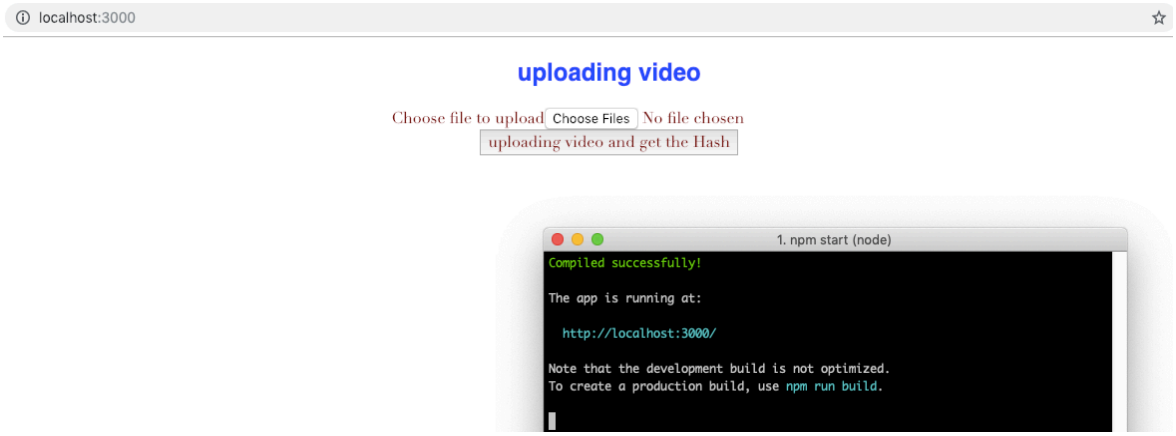

Figure 4.2 Start this Project

Shown as Fig 4.2, we click the “choose files” button to select video files we need to store into the chain. When the selection is over, clicking the “uploading videos and get the hash” button below will upload the video files to the distributed file storage system, a root hash value is generated, which contains all information of the uploaded file. Then we click “store the hash of video on the blockchain”, the system will call the Electrum wallet again as illustrated in Fig 4.3. After the transaction is completed, the video will be saved into the blockchain.

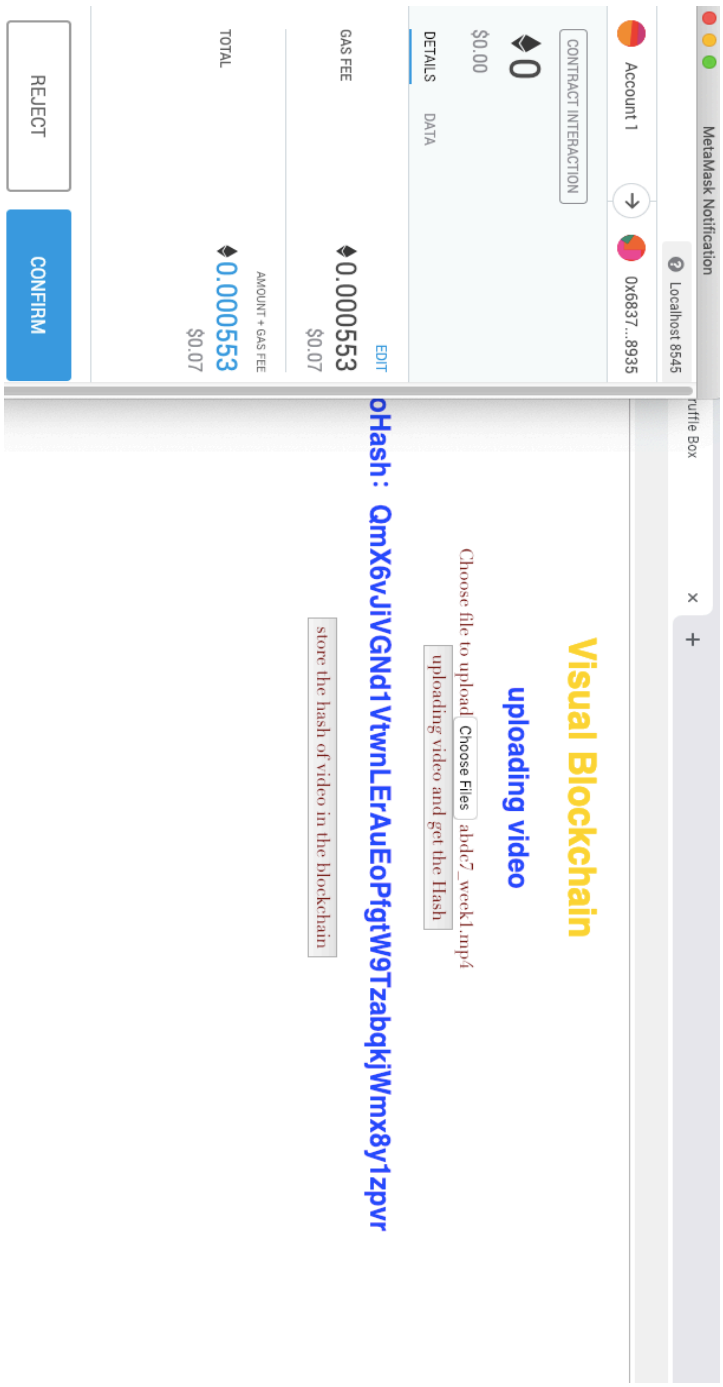

Figure 4.3 Store the Videos on the Blockchain

We read the video by clicking the button “read hash from blockchain”. We can also retrieve the video files stored on the chain, as shown in Fig 4.4.

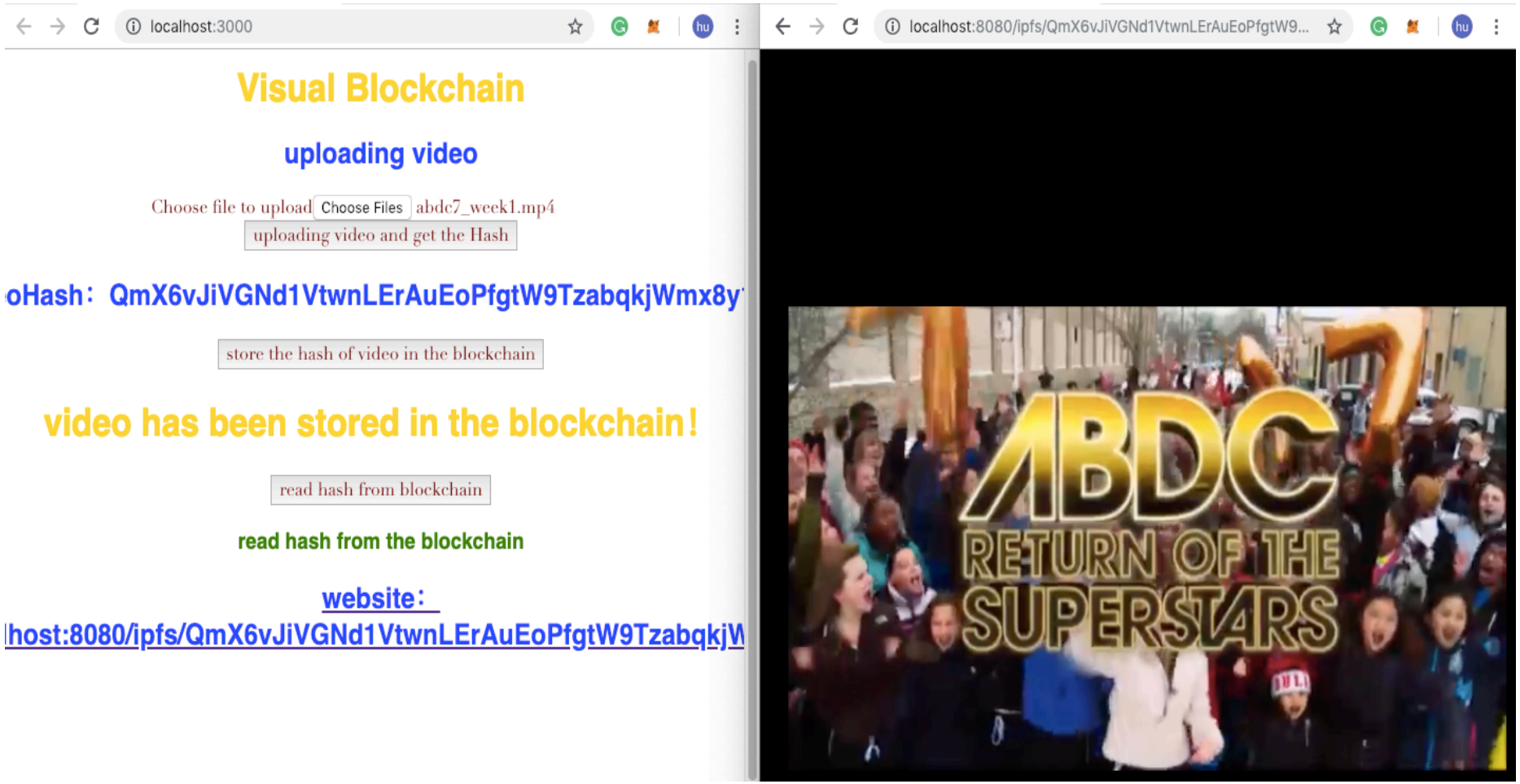

Figure 4.4 Retrieve the Videos Stored in the Blockchain

4.2 Combine Online Video Website with Blockchain

When we search “ABDC7” in the search bar, the relevant results will be shown as Fig 4.5.

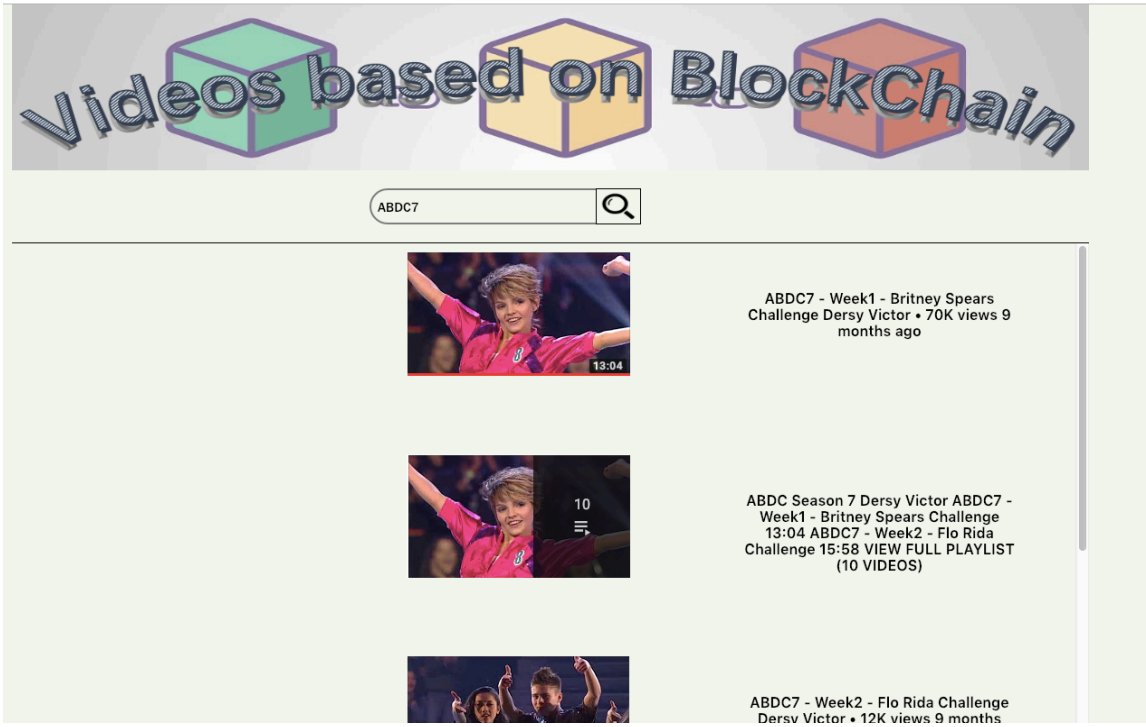

Figure 4.5 Search Interface

Consequently, when we click on the first one in the list, the interface will link to the video playback page. It is evident that the order of the playlist has been sorted.

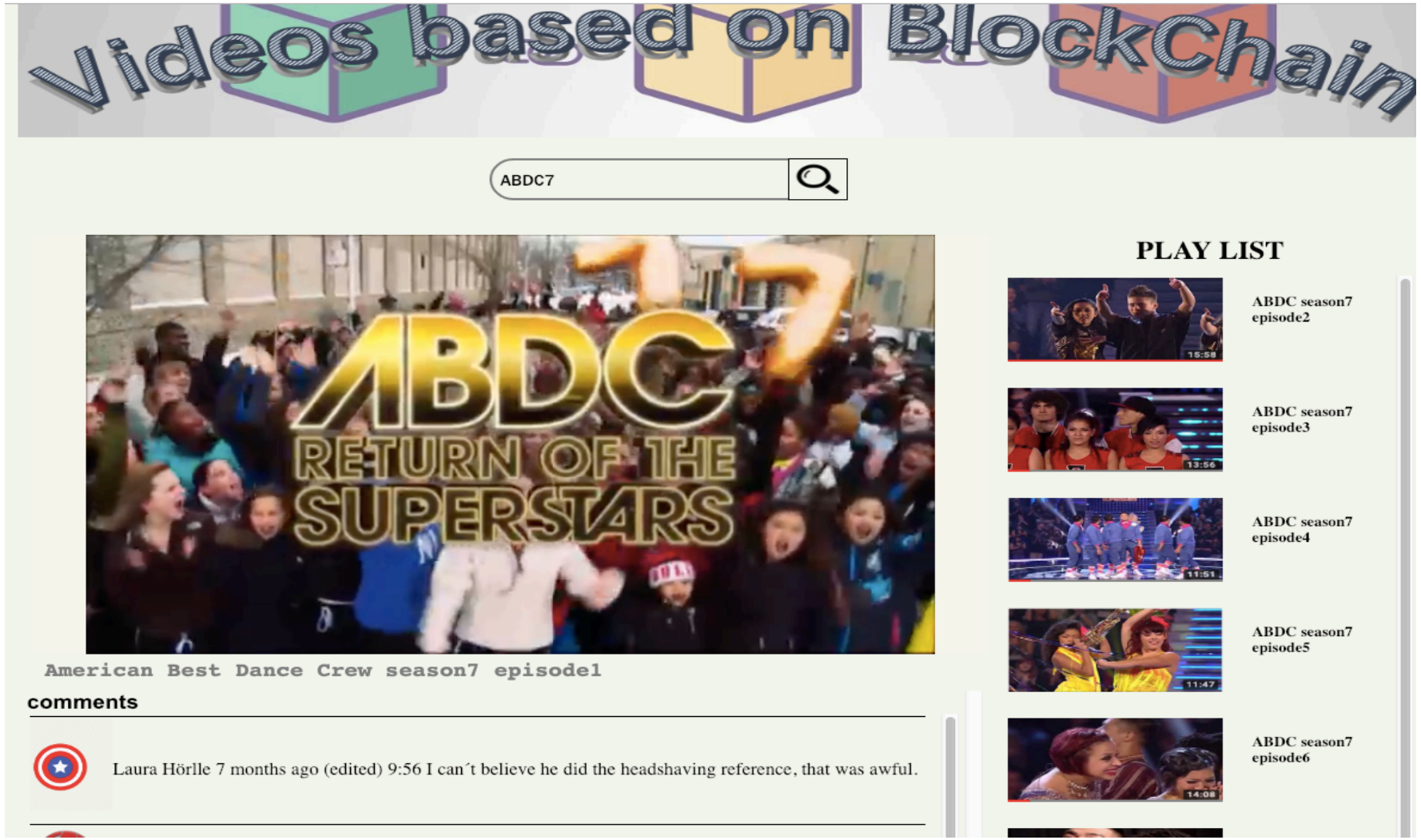

Figure 4.6 The Video Play Page

4.3 The Statistic of This Project

4.3.1 Expenditure

The data store on the blockchain is charged in bytes. In our experiment, the principle of Merkle tree is adopted, the original video data is split into many parts, the divided video data is hashed multiple times, a root hash containing all the data of the video is generated, and finally stored to the chain. We only save the string on the blockchain; thus, we finally store 42-byte string, including the entire video data. There is no cost in executing the transaction itself, as explained in Fig 4.7.

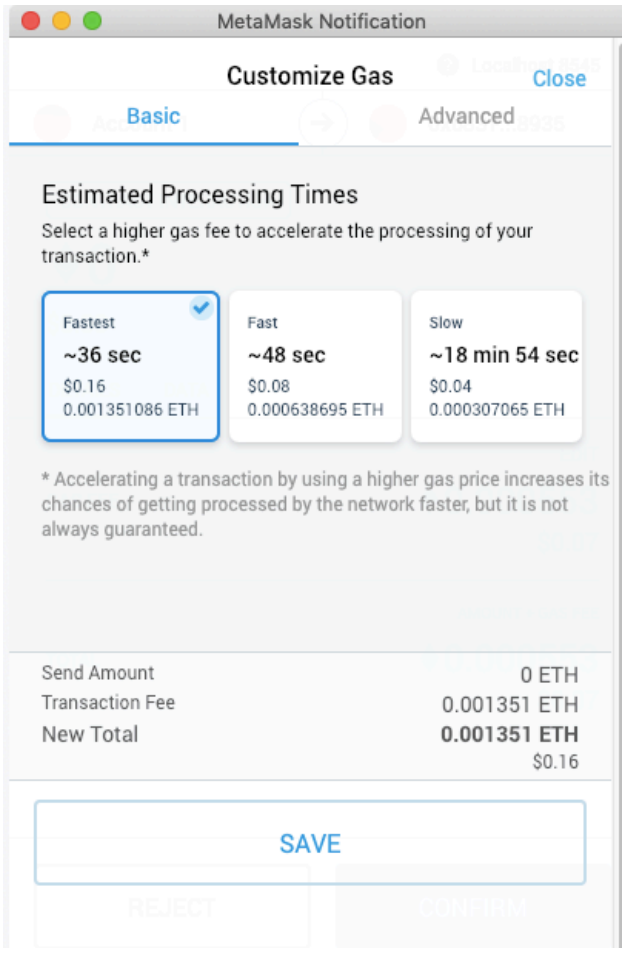

Figure 4.7 Expenditure of Storage

4.3.2 Timeliness

In the world of blockchains, the time it takes for data to be written into the block is determined by the miners, who are one of the crucial maintainers of the system. At the

beginning of development, Bitcoin set up only two ways to obtain Bitcoin: one is mining, the other is to record the transfer transaction and confirm it. With the rise of various digital currencies, this discipline has gradually changed, developers began to occupy a portion of the initially issued currency, but the constant is that recording and confirmation still reward for obtaining digital currency.

The main factor affecting the miners' recording and confirmation is the transaction fee. According to our experiment, the fastest one is less than 36 seconds, and the slow one can be 18 minutes later. This difference is huge, but the cost of both is only 12 points. Therefore, we choose the highest efficiency in this project.

4.3.3 Storage

As we know, in Ethereum system, the storage of each block is only 2Mb, there has been a hard fork expansion to 8Mb. This activity is led by someone who holds many computers which supply more than half computing power in the world. Most of the opponents consider that is contrary to the core technology of the blockchain at the beginning of its invention.

This experiment sufficiently is taken into consideration of the limitations of blockchain capacity with a view related to transaction efficiency issues. As a result of this, we introduce the concept of IPFS, a distributed storage system. The system allows files to be shared on client computers and integrated into the global file system. This technology is based on the BitTorrent protocol and the Distributed Hash Table.

Therefore, anyone can upload any files to the system which is content addressable. Hence, it is impossible to forge a file with a given address. Based on this hypothesis, we ensure that the security of this file is achieved through quickly verifying the BitTorrent protocol. We can freely upload and download the files we need; the data can be confirmed by using the same address. In other words, we can create a database with unlimited storage.

4.3.4 Reordered Playlists

In this section, we utilize blockchain technology to resort online videos; therefore, we assume that all the data on the chain could be sorted in ascending or descending order. In

this experiment, we upload the videos in the order of "American Best Dance Crew season 7 Week 1" to "Week 10", the default order is to store the videos according to the temporary order on the chain.

As shown in Fig 4.8, when the currently playing video is Episode 4, Episode 5 is ready to be played at the first rank in the playlist; the subsequent ones are 6, 7, 8, 9, 10, which give us potential guidance to watch the videos. It provides convenience for us when we watch a TV drama series.

Not only that, when we randomly select an episode in the playlist, in the refreshed webpage, the playlist will be recombined according to the currently playing episode, finally presented in order.

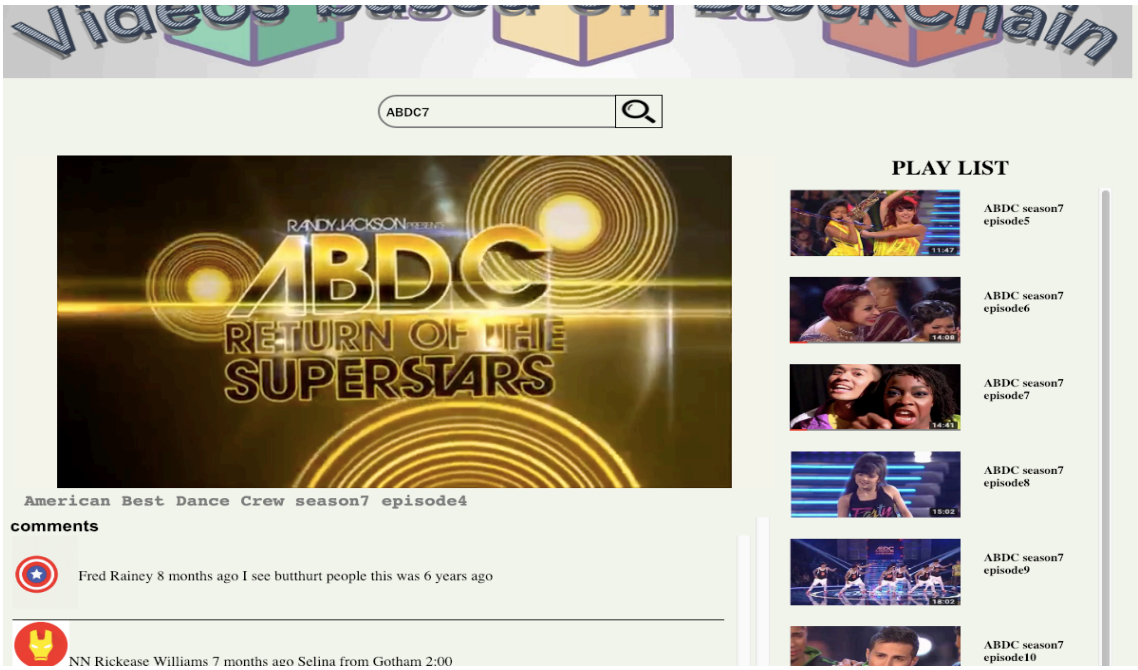

Figure 4.8 The Re-Ordered playlist

4.4 Limitations of the Research

In this project, though combining blockchain and online video play was successfully implemented, the following limitations still exist.

4.4.1 Limitations of Data Diversity

Since the current digital virtual currency is emerging one after another, a new digital currency platform is generated one by one every day. In this experiment, we select Ethereum for our experimentation.

Considered that all creations of digital currencies are based on the development of Bitcoin technology, the core content including the overall framework is basically referenced to the various fundamental technology, e.g., some modified the consensus mechanism, some altered the total amount of the currency, some amended the way to obtain cryptocurrency, and some revised the encryption method, etc. Compared to various types of digital currencies, ETH, which is currently recognized with a Turing-complete development platform, is selected as our experimental basically platform.

4.4.2 Limitations of the Amount of Data

Utilizing blockchain as a space for storing video data for online video websites is a huge gap compared to data processing. Although there is a big gap between the blockchain-based database and the real database, in this experiment, we applied the blockchain structure and attributed to save online videos by using timestamp and assure that the data stored on the blockchain must be arranged in a time sequence. It has also been verified in our experiments.

Chapter 5
Analysis and Discussions

In this chapter, experimental data will be presented and further analysed; especially, the advantages of blockchain applications combined with sorting online videos will be stated by comparing with the traditional online websites.

5.1 Analysis

In this project, we split video data into multiple parts by using the Merkle tree, then hash them, finally get a root hash. Only does the corresponding root hash value need to be stored in the chain, and the remaining original video files are stored in the IPFS distributed storage system. In this way, achieving the goal of storage of big data on the blockchain will not have any negative effect on the efficiency of blockchain. After dealt with the video data by using a Merkle tree, a string having 42 bytes generated by the previous step finally is stored on the blockchain.

Traditional video storage methods are usually separated under the chain or on the chain (Fang, Chen, Wen, & Prybutok, 2018). The method under the chain is divided into local storage and cloud storage. The capacity of local storage depends on the size of local storage space.

Cloud storage also has the problem of resource allocation. It usually costs our resources to purchase the cloud space, but its flexibility is much better than local deposit (Lin, Huang, & Cordie, 2018).

The mode of storage for the chain is a traditional one. The mode of storage on the chain is still relatively unfamiliar to the public, and its technical complexity is higher. Usually, it requires those third party technology to provide corresponding technical support with corresponding smart contracts or to simplify storage on the chain. The cost of saving the same video data is much higher than usual (Liu, Suh, & Wagner, 2016).

We set the first episode of videos to store on the chain as an example. According to this experiment, in Fig 4.7, the cost for saving a 66.76Mb video needs only 16 cents.

On the other hand, we provide blockchain as a database for online video sites. We could see the re-ordered playlist located in the next zone.

Concerning the traditional online video search, we take YouTube as an example. When we search for a video, the search result will be listed as the next episode or some other random and relevant videos in the playlist. That is owing to the mechanism of YouTube search.

If we enter “ABDC7” on the website of YouTube, the relevant search items will be listed randomly. Because we have not seen this video before, we want to start from the first episode; if we click on the first video in the playlist, then it updates the interface with a new one as shown in Fig. 5.2; apparently, it does not have the next episode in the list to be played; therefore, the user must go back to the previous page, as shown in Fig. 5.1.

Figure 5.1 The Result after Search “ABDC7” between YouTube and Our Experiment

Figure 5.2 The Playing Page in YouTube

We compare the traditional search results to get the full episodes of the video playlist with our proposed method. We will measure the distance between the two playlists with or without re-sorting. As shown in Fig 5.3, it is a playlist about the full episodes on the YouTube website.

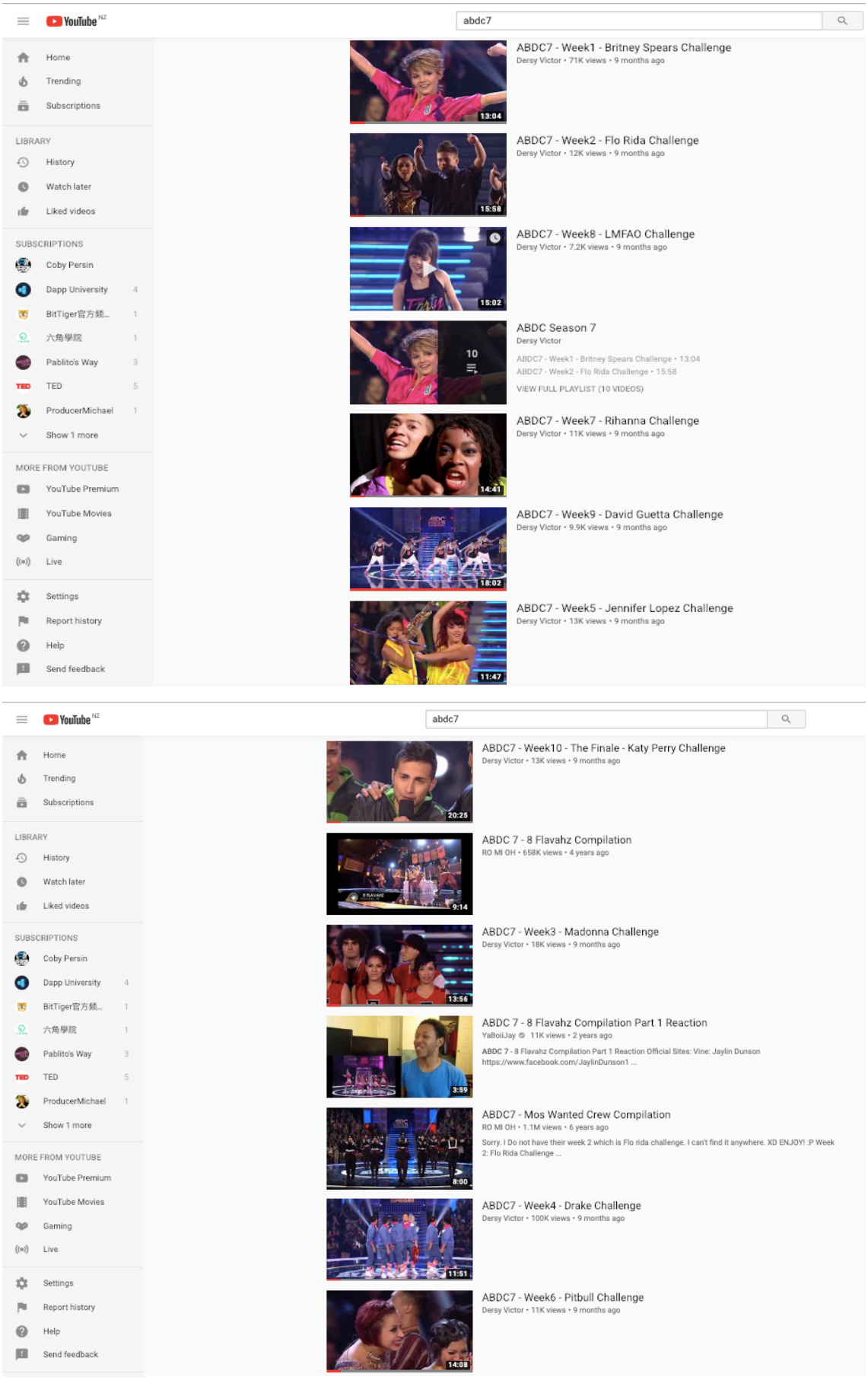

Figure 5.3 The Playlist Containing Full Episodes Information on YouTube

Figure 5.4 The Playlist Containing Full Episodes Information in Our Experiment

5.2 Discussion

We expect the searched videos could be sorted in ascending order, and it could be accomplished through our experiment; as Fig 5.4 indicates, the items in the playlist arranged in a specific order automatically when we click a link associated with an episode. However, some related contents or others always appear in the playlist zone when entering the YouTube interface, we must return to the previous page to search what we want, it is a waste process. Thus, we directly use the previous searching page as the comparing object, ignoring the wasted time of switching back and forth between interfaces.

We simplify our search results, as shown in Table 5.1. The left list demonstrates the playlist on YouTube, the middle list is related to the target, and the right list is the playlist generated from our experiment, the results are listed according to ascending or descending order. We use a box containing a number on the right side of the name of the episodes as

a logo. Meanwhile, we find that some mismatched items in the left list, which are marked by using character “X”.

Table5.1 The Simplified Sequences

	YouTube		Target		Experiment	
row1	ABDC7 week1	1	ABDC7 week1	1	ABDC7 week1	1
row2	ABDC7 week2	2	ABDC7 week2	2	ABDC7 week2	2
row3	ABDC7 week8	8	ABDC7 week3	3	ABDC7 week3	3
row4	distracters	X	ABDC7 week4	4	ABDC7 week4	4
row5	ABDC7 week7	7	ABDC7 week5	5	ABDC7 week5	5
row6	ABDC7 week9	9	ABDC7 week6	6	ABDC7 week6	6
row7	ABDC7 week5	5	ABDC7 week7	7	ABDC7 week7	7
row8	ABDC7 week10	10	ABDC7 week8	8	ABDC7 week8	8
row9	distracters	X	ABDC7 week9	9	ABDC7 week9	9
row10	ABDC7 week3	3	ABDC7 week10	10	ABDC7 week10	10
	distracters	X				
	distracters	X				
	ABDC7 week4	4				
	ABDC7 week6	6				

In order to calculate the distance between the two lists, we introduce the “edit distance” which may also be referred to Levenshtein distance (Yujian & Bo, 2007; Runkler & Bezdek, 2000; Kumar, Agarwal, & Bhagvati, 2014; Doran & van Wamelen, 2010) as shown in Eq (5.1).

$$lev_{a,b}(\mathrm{i,j}) = \begin{cases} \max(i,j) & if\ \min(i,j) = 0, \\ min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1(a_i \neq b_j) \end{cases} & otherwise. \end{cases} \quad (5.1)$$

We convert Eq. (5.1) into a matrix and make it more intuitive as indicated in Table 5.2 (Wachter-Zeh & Antonia, 2018). The first row represents the YouTube list, the first column stands for the target list. We follow the following steps to get the result.

Step 1. A matrix is initialized.

Step 2. The matrix could be filled obeying the rules from the left row to the right row and from top to bottom.

Step 3. Each horizontal or vertical cell corresponding to an insertion or a deletion, respectively.

Step 4. The cost of each cell is set to 1.

Step 5. Each cell is filled with the minimum value. If the two video positions are same, the cell will be 1, else 0.

Step 6. The number in the lower right corner in the matrix is the Levenshtein distance between two objects.

Table 5.2 The Matrix Used to Calculate the Edit Distance

Search for "ABDC7"		YouTube List													
		1	2	8	X	7	9	5	10	X	3	X	X	4	6
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
2	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12
3	3	2	1	1	2	3	4	5	6	7	7	8	9	10	11
4	4	3	2	2	2	3	4	5	6	7	8	8	9	9	10
5	5	4	3	3	3	3	4	4	5	6	7	8	9	10	10
6	6	5	4	4	4	4	4	5	5	6	7	8	9	10	10
7	7	6	5	5	5	4	5	5	6	6	7	8	9	10	11
8	8	7	6	5	6	5	5	6	6	7	7	8	9	10	11
9	9	8	7	6	6	6	5	6	7	7	8	8	9	10	11
10	10	9	8	7	7	7	6	6	6	7	8	9	9	10	11

Target List

The number at the lower right corner is 11, which means the distance between the YouTube list and Target list is 11. If the edit distance between the test list and the target list is lower, the similarity between them is higher, so it is obvious that our experimental data is closer to the target, because they are the same, and the distance between them is zero.

We searched several keywords while applying this method, we simplify them as follows:

When we search for "ABDC2" both in YouTube and our experiment, Table 5.3 describes the result of that.

Table 5.3 Search for “ABDC2”

YouTube		Target	Experiment
ABDC2 episode1	1	1	1
ABDC2 episode7	7	2	2
ABDC2 episode2	2	3	3
distracters	X	4	4
ABDC2 episode3	3	5	5
ABDC2 episode6	6	6	6
ABDC2 episode5	5	7	7
ABDC2 episode9	9	8	8
ABDC2 episode4	4	9	9
ABDC2 episode10	10	10	10
distracters	X		
distracters	X		
distracters	X		
ABDC2 episode8	8		

The distance between YouTube list and Target list is 10 as shown in Table 5.4.

Table5.4 The Matrix Used to Calculate the Edit Distance

Search for “ABDC2”		YouTube List													
		1	7	2	X	3	6	5	9	4	10	X	X	X	8
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
2	2	1	1	1	2	3	4	5	6	7	8	9	10	11	12
3	3	2	2	2	2	2	3	4	5	6	7	8	9	10	11
4	4	3	3	3	3	3	3	4	5	5	6	7	8	9	10
5	5	4	4	4	4	4	4	3	4	5	6	7	8	9	10
6	6	5	5	5	5	5	4	4	4	5	6	7	8	9	10
7	7	6	5	6	6	6	5	5	5	5	6	7	8	9	10
8	8	7	6	7	7	7	6	6	6	6	6	7	8	9	9
9	9	8	7	8	8	8	7	7	6	7	7	7	8	9	10
10	10	9	8	9	9	9	8	8	7	8	7	8	8	9	10

Target List

When we change the key words to “Uncle Season1” the results show as Table 5.5.

Table 5.5 Search for “Uncle Season1”

YouTube		Target	Experiment
Uncle S1E1	1	1	1
Uncle S1E2	2	2	2
Uncle S1E3	3	3	3
Uncle S1E5	5	4	4
distracters	X	5	5
distracters	X	6	6
Uncle S1E4	4		
Uncle S1E6	6		

The distance between YouTube list and Target list is 4 shown in Table 5.6.

Table 5.6 The Matrix after Calculated the Edit Distance

Search for “Uncle Season1”		YouTube List							
		1	2	3	5	X	X	4	6
	0	1	2	3	4	5	6	7	8
1	1	0	1	2	3	4	5	6	7
2	2	1	0	1	2	3	4	5	6
3	3	2	1	0	1	2	3	4	5
4	4	3	2	1	1	2	3	3	4
5	5	4	3	2	1	2	3	4	4
6	6	5	4	3	2	2	3	4	4

Target List

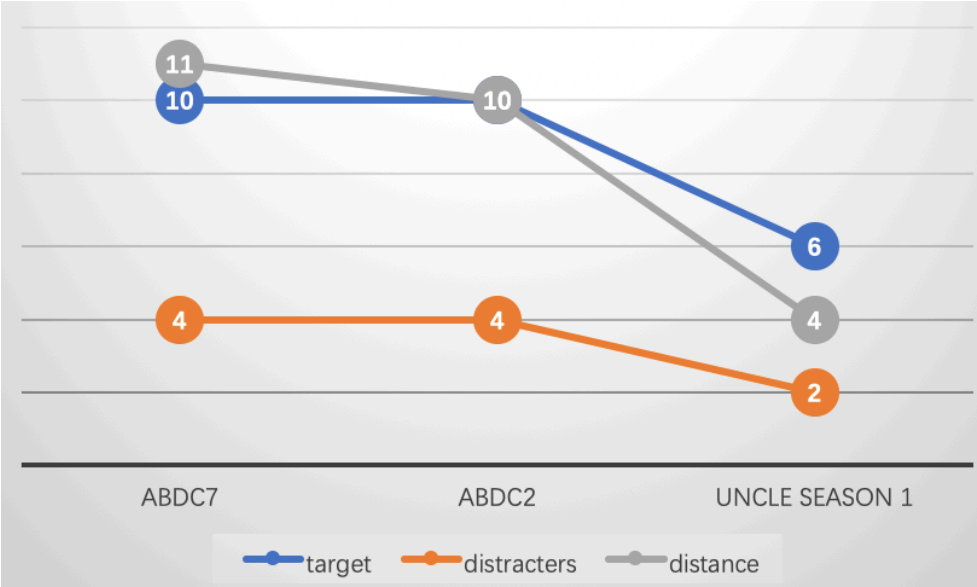

Figure 5.5 Results Comparison

In Fig 5.5, the number of distracters is reduced as the number of target decreases, and the distance also presents towards a downtrend.

Through comparing the results, it is evident that not only the cost of big data uplink storage is saved, but also the transaction time of the winding is stored in our experiments.

We utilized blockchain as a database to ensure the robustness and immutability of video data; on the one hand, the playlists are sorted in the right order, which saves the retrieval time of video browsing and improves the efficiency of video sites.

Chapter 6
Conclusion and Future Work

We will make a summary about our experiments in this chapter, we will also vision future research work according to the results we generated as well as limitations of this thesis project.

6.1 Conclusion

Nowadays, our entire communities pay attention to blockchain technology (Weber, 2018). More research institutions are working toward improving the infrastructure of blockchain and reducing the entry barrier of blockchain. One of the facts of blockchain is that more and more applications are invented; they are used to our daily lives (Halaburda, 2018).

However, how to store big data is always an issue to perplex all the researchers and developers in this field, because the primary problem is that it is easy to resolve this matter of blockchain by increasing the size of each block, it will bring new problems; additionally, the actors lose their enthusiasm with the lower timeliness (Filipova, 2018).

This thesis attempts to solve the problem of saving big videos in blockchain while not affecting the running speed and efficiency of the entire chain. Using the existing Merkle tree to the blockchain, the video data to be saved is stored with a 42 bytes hash value. With the utilization of distributed file management system (IPFS), we preserve the original video data in it so that we can download or upload them when we need to call this file in future; we only need to keep the 42 bytes hash value which can lead to the address in the IPFS system.

Besides, we found that more and more people focus on online videos either through PC or mobile devices, this occupies their most spare time. But the playlist of the episodes in an online video website is always not sorted in right order, which will waste users' time when they browse these websites and find the videos.

We consider the data stored in each block is resorted in chronological order by using the timestamp feature. With this feature, firstly, we have created a private chain that used as a database to store video data; then, another function of this database stores videos in the blockchain by using timestamps. Combining these features leads the playlist to be organized in an ascending or descending order; we thus save much time when we browser these online websites with the help of visual blockchain. Another feature is that the updated playlist is still in order, and the next episode is adjacent to the playing one at present when we click on any of the playlists.

6.2 Future Work

Although we have obtained the expected experimental results in these experiments; however, we still have some to be further improved and strengthened in future.

In our experiment, we created a private blockchain locally, and all the data is based on the private network. Our experiment has achieved the expected results based on theoretical research or in an ideal environment. In future, we will try to connect to the main network and synchronize the corresponding video data.

We only created one private blockchain in this project, and all subsequent data is stored on the blockchain. Our outlook can construct multiple blockchains at the same time. They only directly connect each other to achieve multi-dimensional linkages of data and finally build a crisscross blockchain network. We hope that each chain will have its main functions in future.

We also consider combined blockchains with video surveillance and deep learning. The videos captured on the road are saved in a blockchain and sorted according to its timestamps. When a suspicious car needs to be tracked, we identify its license plate and appearance. Sorting image data by using blockchain is to generate a time sequence of video frames, which can be helpful to track traffic criminals. Figure 6.1 described the combination of visual blockchain and surveillance, the videos are stored on the blockchain sequentially from Clip 1 to Clip 12, which cannot be tampered.

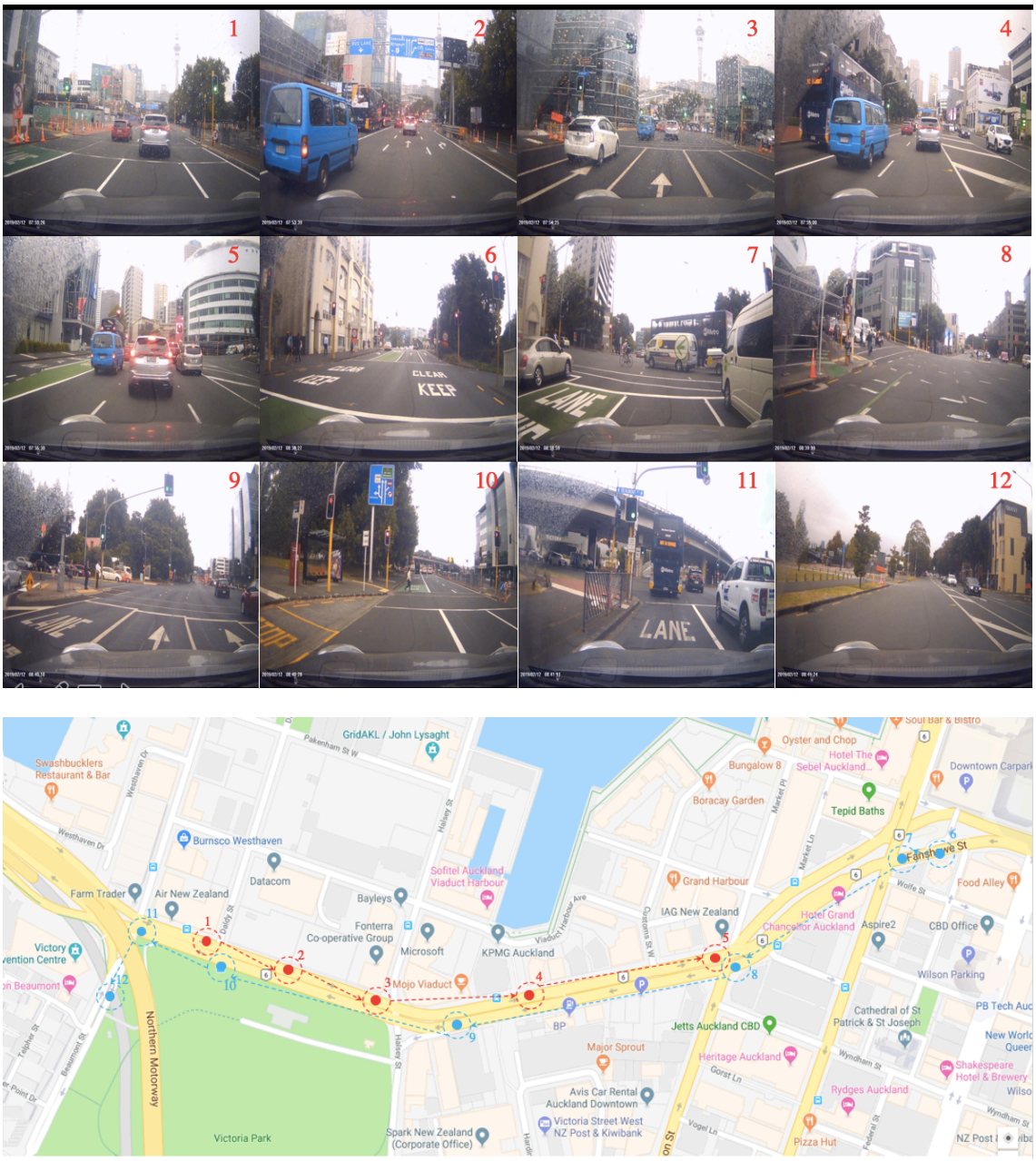

Figure 6.1 Visual Blockchain Applied to Surveillance Scenarios

References

Alcarria, R., Bordel, B., Robles, T., Martín, D., & Manso-Callejo, M. Á. (2018). A blockchain-based authorization system for trustworthy resource monitoring and trading in smart communities. *Sensors (14248220), 18(10), pp.3561.*

Alessi, M., Camillo, A., Giangreco, E., Matera, M., Pino, S., & Storelli, D. (2018). Make Users Own Their Data: A Decentralized Personal Data Store Prototype Based on Ethereum and IPFS. *3rd International Conference on Smart and Sustainable Technologies (SpliTech) Smart and Sustainable Technologies (SpliTech).*

Andreeva, E., Bouillaguet, C., Dunkelman, O., Fouque, P., Hoch, J., Kelsey, J., Shamir, A., & Zimmer, S. (2016). New second-preimage attacks on hash functions. *Journal of Cryptology, 29(4), pp.657-696.*

Ao, L., Cruickshank, H., Yue, C., Asuquo, P., Anyigor Ogah, C. P., & Zhili, S. (2017). Blockchain-Based dynamic key management for heterogeneous intelligent transportation systems. *Internet of Things Journal, IEEE, 4(6), pp.1832-1843.*

Appelbaum, D., & Stein Smith, S. (2018). Blockchain Basics and Hands-on Guidance: Taking the Next Step toward Implementation and Adoption. *CPA Journal, 88(6), pp.28-37.*

Aste, T., Tasca, P., & Di Matteo, T. (2017). Blockchain Technologies: The Foreseeable Impact on Society and Industry. *Computer (00189162), 50(9), pp.18-28.*

Badra, M., & Borghol, R. (2018). Long-term integrity and non-repudiation protocol for multiple entities. Sustainable cities and society*, 40, pp.189-193.*

Balajee, M. (2018). Bitcoin Generation using Blockchain Technology. *JOIV: International Journal on Informatics Visualization, 2(3), pp.127-132.*

Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2014). Sufficient conditions for sound tree and sequential hashing modes. *International Journal of Information Security, 13(4), pp.335-353.*

BitCoin technology Merkle tree (Hash tree). (2016).

Bock, M. A. (2016). Showing versus telling: Comparing online video from newspaper and television websites. Journalism*,17(4), pp.493-510.*

Chen, L., Xu, L., Gao, Z., Lu, Y., & Shi, W. (2018). Tyranny of the Majority: On the (Im)possibility of Correctness of Smart Contracts. *Privacy Security & Privacy, IEEE, 16(4), pp.30-37.*

Chen, Y., Ding, S., Zheng, H., Yang, S., & Xu, Z. (2018). Blockchain-Based Medical Records Secure Storage and Medical Service Framework. *Journal of Medical Systems, 43(1).*

Chen, Y., Li, H., Li, K., & Zhang, J. (2017). An improved P2P file system scheme based on IPFS and Blockchain. *IEEE International Conference on, pp.2652-2657.*

Cho, H. (2018). ASIC-Resistance of Multi-Hash Proof-of-Work Mechanisms for Blockchain Consensus Protocols. *IEEE Access, 6, pp.66210-66222.*

Cho, S.H., Park, S.Y., & Lee, S.R. (2017). Blockchain consensus rule based dynamic blind voting for non-dependency transaction. *International Journal of Grid and Distributed Computing,10(12), pp.93-106.*

Dai, M., Zhang, S., Wang, H., & Jin, S. (2018). A Low Storage Room Requirement Framework for Distributed Ledger in Blockchain. *IEEE Access,6, pp.22970-22975.*

Davidson, L., & Block, W. E. (2015). Bitcoin, the regression theorem, and the emergence of a new medium of exchange. *Quarterly Journal of Austrian Economics. 18(3), pp.311-338.*

Doran, H. C., & van Wamelen, P. B. (2010). Application of the Levenshtein distance metric for the construction of longitudinal data files. *Educational Measurement: Issues and Practice, 29(2), pp.13-23.*

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018). Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge & Data Engineering, 30(7), pp.1366-1385.*

Fairley, P. (2017). Blockchain world - Feeding the blockchain beast if bitcoin ever does go mainstream, the electricity needed to sustain it will be enormous. *Spectrum, IEEE, 54(10), pp.36-59.*

Fang, J., Chen, L., Wen, C., & Prybutok, V. R. (2018). Co-viewing Experience in Video Websites: The Effect of Social Presence on E-Loyalty. *International Journal of Electronic Commerce, 22(3), pp.446-476.*

Filipova, N. (2018). Blockchain – An opportunity for developing new business models. *Business Management / Biznes Upravlenie, (2), pp.75-92.*

Freund, A. (2017). Economic incentives and Blockchain security. *Journal of Securities Operations & Custody, 10(1), pp.67-76.*

Gazi, P., Kiayias, A., & Russell, A. (2018). Stake-Bleeding Attacks on Proof-of-Stake Blockchains. *Crypto Valley Conference on Blockchain Technology (CVCBT), pp.85-92.*

Genkin, D., Papadopoulos, D., & Papamanthou, C. (2018). Privacy in decentralized cryptocurrencies. *Communications of the ACM, 61(6), pp.78-88.*

Gouru, N., & Vadlamani, N. L. (2018). CoPS - Cooperative provenance system with zkp using ethereum blockchain smart contracts. *International Journal of Distributed Systems and Technologies, 9(4), pp.40-53.*

Gramoli, V., & Staples, M. (2018). Blockchain Standard: Can We Reach Consensus? *Communications Standards Magazine, IEEE, 2(3), pp.16-21.*

Halaburda, H. (2018). Economic and Business Dimensions Blockchain Revolution without the Blockchain? *Communications of the ACM, 61(7), pp.27-29.*

Hasan, M. R., Jha, A. K., & Liu, Y. (2018). Excessive use of online video streaming services: Impact of recommender system use, psychological factors, and motives. *Computers in Human Behavior, 80, pp.220-228.*

Hass, M. (2014). Under the microscope: Economic and Environmental cost of bitcoin mining.

Hawlitschek, Florian., Notheisen, B., & Teubner, T. (2018). The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy. *Electronic Commerce Research & Applications, 29, pp.50-63.*

Herlihy, M. (2019). Blockchains from a Distributed Computing Perspective. *Communications of the ACM,* 62(2), *pp.78-85.*

Hongwei, L., Rongxing, L., Liang, Z., Bo, Y., & Xuemin, S. (2014). An Efficient Merkle-Tree-Based Authentication Scheme for Smart Grid. *IEEE Systems Journal, 8(2), pp.655-663.*

Hu, B., & Gharavi, H. (2014). Smart Grid Mesh Network Security Using Dynamic Key Distribution With Merkle Tree 4-Way Handshaking. *IEEE Transactions on Smart Grid, 5(2), pp.550-558.*

Hu, Y., Xiong, Y., Huang, W., & Bao, X. (2018). KeyChain: Blockchain-Based Key Distribution. *International Conference on Big Data Computing and Communications (BIGCOM) pp.126-131.*

Jan, K., & Geir, H. (2018). Tail Removal Block Validation: Implementation and Analysis. *Modeling, Identification and Control, 39(3), pp.151-156.*

Jesus, E. F., Chicarino, V. R. L., de Albuquerque, C. V. N., & Rocha, Antônio A. de A. (2018). A Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack. *Security & Communication Networks, pp.1-27.*

Ji, Y., Zhang, J., Ma, J., Yang, C., & Yao, X. (2018). BMPLS: Blockchain-Based Multi-level Privacy-Preserving Location Sharing Scheme for Telecare Medical Information Systems. *Journal of Medical Systems, 42(8).*

Keenan, T. P. (2017). Alice in Blockchains: Surprising Security Pitfalls in PoW and PoS Blockchain Systems. *Annual Conference on Privacy, Security and Trust (PST), pp.400-4002.*

Kewell, B., Adams, R., & Parry, G. (2017). Blockchain for good? *Strategic Change, 26(5), pp.429-437*.

Kewell, B.& Michael Ward, P. (2017). Blockchain futures: With or without Bitcoin? *Strategic Change, 26(5), pp.491-498.*

Kieran, D., Yan, W. (2010) A framework for an event-driven video surveillance system. IEEE AVSS, Boston, USA.

Kokina, J., Mancha, R., & Pachamanova, D. (2017). Blockchain: Emergent Industry Adoption and Implications for Accounting. *Journal of Emerging Technologies in Accounting, 14(2), pp.91-100.*

Koo, D., Shin, Y., Yun, J., & Hur, J. (2018). Improving Security and Reliability in Merkle Tree-Based Online Data Authentication with Leakage Resilience. *Applied Sciences (2076-3417), 8(12), pp.2532.*

Kugler, L. (2018). Why cryptocurrencies use so much energy--and what to do about it. *Communications of the ACM, 61(7), pp.15-17.*

Kumar, P. P., Agarwal, A., & Bhagvati, C. (2014). A string matching based algorithm for performance evaluation of mathematical expression recognition. Sadhana-Academy proceedings in engineering sciences, *39 (1), pp.63-79.*

Lacity, M. C. (2018). Addressing Key Challenges to Making Enterprise Blockchain Applications a Reality. *MIS Quarterly Executive, 17(3), pp.201-222.*

Lin, D., & Tang, Y. (2018). Blockchain Consensus Based User Access Strategies in D2D Networks for Data-Intensive Applications. *IEEE Access, 6, pp.72683-72690.*

Lin, X., Huang, M., & Cordie, L. (2018). An Exploratory Study: Using Danmaku in Online Video-Based Lectures. *Educational Media International, 55(3), pp.273-286.*

Liu, L., Suh, A., & Wagner, C. (2016). Watching online videos interactively: the impact of media capabilities in Chinese Danmaku video sites. *Chinese Journal of Communication, 9(3), pp.283-303.*

Liu, M. Z., & Zou, Z. (2018). The application of block chain technology in spot exchange. *Journal of Intelligent & Fuzzy Systems, 34(2), pp.985-993.*

Liu, C., Ranjan, R., Yang, C., Zhang, X., Wang, L., & Chen, J. (2015). MuR-DPA: top-down levelled multi-replica Merkle hash tree based secure public auditing for dynamic big data storage on cloud. *IEEE Transactions on Computers, 64(9), pp.2609-2622.*

Li, J., Wu, J., & Chen, L. (2018). Block-secure: Blockchain based scheme for secure P2P cloud storage. *Information Sciences, 465, pp.219-231*

Li, M. N. (2017). Analyzing Intellectual Structure of Related Topics to Blockchain and Bitcoin: From Co-citation Clustering and Bibliographic Coupling Perspectives. *Zidonghua Xuebao/Acta Automatica Sinica, 43(9), pp.1509-1519*

Li, R., Song, T., Mei, B., Li, H., Cheng, X. & Sun, L. (2018). Blockchain For Large-Scale Internet of Things Data Storage and Protection. *IEEE Transactions on Services Computing, In Press.*

Lu, Y. (2018). Blockchain and the related issues: a review of current research topics. *Journal of Management Analytics, 5(4), pp.231-255.*

Ma, Z., Huang, W., Bi, W., Gao, H., & Wang, Z. (2018). A master-slave blockchain paradigm and application in digital rights management. *Communications, China, 15(8), pp.174-188.*

Mahdi, H. M., & Maaruf, A. (2018). Applications of Blockchain Technology beyond Cryptocurrency. *Annals of Emerging Technologies in Computing, 2(1), pp.1-6.*

Manski, S. (2017). Building the blockchain world: Technological commonwealth or just more of the same? *Strategic Change, 26(5), pp.511-522.*

Mao, J., Zhang, Y., Li, P., Li, T., Wu, Q., & Liu, J. (2017). A position-aware Merkle tree for dynamic cloud data integrity verification. *Soft Computing - A Fusion of Foundations, Methodologies & Applications, 21(8), pp.2151-2164.*

Maxim, Y. A., Anastasiya, A. K., Sergey, A. S., Yuri, V. F., & Anastasiia, S. S. (2018). A Design of Cyber-physical Production System Prototype Based on an Ethereum Private Network. *The Conference of Open Innovations Association FRUCT, 426(22), pp.3-11.*

Mehar, M. I., Shier, C. L., Giambattista, A., Gong, E., Fletcher, G., Sanayhie, R., Kim, H. M., & Laskowski, M. (2019). Understanding a Revolutionary and Flawed Grand Experiment in Blockchain: The DAO Attack. *Journal of Cases on Information Technology, 21(1), pp.19-32.*

Meloni, A., Madanapalli, S., Divakaran, S. K., Browdy, S. F., Paranthaman, A., Jasti, A., Krishna, N., & Kumar, D. (2018). Exploiting the IoT Potential of Blockchain in the IEEE P1931.1 ROOF Standard. *IEEE Communications Standards Magazine,2(3), pp.38-44.*

Munoz, J. L., Forne, J., Esparza, O., & Soriano, M. (2004). Certificate revocation system implementation based on the Merkle hash tree. *International Journal of Information Security, 2(2), pp.110-124.*

Neyer, G., & Geva, B. (2017). Blockchain and payment systems: What are the benefits and costs? *Journal of Payments Strategy & Systems, pp.215-225.*

O'Dair, M., & Beaven, Z. (2017). The networked record industry: How blockchain technology could transform the record industry. *Strategic Change, 26(5), pp.471-480.*

O'Leary, D. E. (2018). Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems. *Intelligent Systems in Accounting, Finance & Management, 24(4), pp.138-147.*

Pãnescu, A., & Manta, V. (2018). Smart Contracts for Research Data Rights Management over the Ethereum Blockchain Network. *Science & Technology Libraries, 37(3), pp.235-245.*

Paul, C., Jay, A., & Emre, S. (2016). Deep Neural Networks for YouTube Recommendations. *https://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/45530.pdf.*

Pierro, M. D. (2017) What Is the Blockchain? *Computing in Science & Engineering, 19(5), pp.92-95.*

Ramezan, G., & Leung, C. (2018). A Blockchain-Based Contractual Routing Protocol for the Internet of Things Using Smart Contracts. *Wireless Communications & Mobile Computing, pp.1-14.*

Ren, Y., Liu, Y., Ji, S., Sangaiah, A. K., & Wang, J. (2018). Incentive Mechanism of Data Storage Based on Blockchain for Wireless Sensor Networks. *Mobile Information Systems, pp.10.*

Risius, M., & Spohrer, K. (2017). A Blockchain Research Framework. *Business & Information Systems Engineering, 59(6), pp.385-409.*

Rosa, R., & Rothenberg, C. E. (2018). Blockchain-Based decentralized applications for multiple administrative domain networking. *Communications Standards, IEEE, 2(3), pp.29-37.*

Runkler, T. A., & Bezdek, J. C. (2000). Automatic keyword extraction with relational clustering and Levenshtein distances. *IEEE International Conference on Fuzzy Systems, 2, pp.636-640.*

Saberi, S., Kouhizadeh, M., & Sarkis, J. (2018). Blockchain technology: A panacea or pariah for resources conservation and recycling? *Resources, Conservation & Recycling, 130, pp.80-81.*

Sehra, A., Cohen, R., & Arulchandran, V. (2018). On cryptocurrencies, digital assets and private money. *Journal of Payments Strategy & Systems,12(1), pp.13-32.*

Shermin, V. (2017). Disrupting governance with blockchains and smart contracts. *Strategic Change, 26(5), pp.499-509.*

Shiyong, Y., Jinsong, B., Yiming, Z., & Xiaodi, H. (2017). M2M Security Technology of CPS Based on Blockchains. *Symmetry (20738994), 9 (9), p193.*

Siba, T. K., & Prakash, A. (2016). Block-Chain: An Evolving Technology. *Global Journal of Enterprise Information System, 8(4), pp.29-35.*

Sifah, E. B., Agyekum, K. O. O., Amofa, S., Xia, Q., Gao, J., Chen, R., Xia, H., Gee, J. C., Du, X., & Guizani, M. (2018). Chain-based big data access control infrastructure. *Journal of Supercomputing, 74(10), pp.4945-4964.*

SOMPOLINSKY, Y., & ZOHAR, A. (2018). Bitcoin's Underlying Incentives. *Communications of the ACM, 61(3), pp.45-53.*

Stavrou, A., & Voas, J. (2017). Verified Time. *Computer (00189162), 50(3), pp.78-82.*

Stewart, I., Ilie, D., Zamyatin, A., Werner, S., Torshizi, M. F., & Knottenbelt, W. J. (2018). Committing to quantum resistance: A slow defence for Bitcoin against a fast quantum computing attack. *Royal Society Open Science, 5(6).*

Shu, Y. (2018) Blockchain for security of a cloud-based online auction system. *Master's thesis, Auckland University of Technology, New Zealand.*

Shu, Y. Yu, J. Yan, W. (2019) Blockchain for Security of Cloud-Based Online Auction. *Exploring Security in Software Architecture and Design, pp.189-210.*

Shu, Y. Yu, J. Yan, W. (2019) State Actor Model for Cloud-Based Online Auction. *Exploring Security in Software Architecture and Design, pp. 170-188*

Suárez-Albela, M., Fraga-Lamas, P., & Fernández-Caramés, T. M. (2018). A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices. *Sensors (14248220), 18(11), pp.3868.*

Suat ÖZDEMİR. (2009). False Data Detection in Wireless Sensor Networks via Merkle Hash Trees. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 11(3).*

Subramanian, H. (2018). Decentralized Blockchain-Based Electronic Marketplaces. *Communications of the ACM, 61(1), pp.78-84.*

Tan, X., Guo, Y., Chen, Y., & Zhu, W. (2018). Accurate inference of user popularity preference in a large-scale online video streaming system. *Science China, Information Sciences, 61(1).*

Tang, H., Shi, Y., & Dong, P. (2019). Public blockchain evaluation using entropy and TOPSIS. *Expert Systems with Applications, 117, pp.204-210.*

Tsukerman, M. (2015). The block is hot: A survey of the state of bitcoin regulation and suggestions for the future. *Berkeley Technology Law Journal, (30), pp.1127-1169.*

Thombs, M., & Tillman, A. A. (2018). Designing 21st Century Curriculum for Bitcoin and Blockchain Studies. *International Journal of Global Business, 11(1), pp.67-80.*

Tschorsch, F. & Scheuermann, B. (2016). Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *Communications Surveys & Tutorials, IEEE, 18(3), pp.2084-2123.*

Underwood, S. (2016). Blockchain Beyond Bitcoin. *Communications of the ACM, 59(11), pp.15-17.*

Vladimir, I. B., & Dmitry, A. M. (2018). On the issue of the blockchain technology perspectives. The more things change, the more they stay the same. *Bezopasnost' Informacionnyh Tehnologij, 25(4), pp.23-33.*

Wachter-Zeh, & Antonia. (2018). List Decoding of Insertions and Deletions. *IEEE Transactions on Information Theory, 64(9), pp.6297-6304.*

Wang, E., Yan, W. (2014) iNavigation: an image-based indoor navigation system. *Multimedia tools and applications, 73 (3), pp.1597-1615*

Wang, H., He, D., & Ji, Y. (2017). Designated-verifier proof of assets for bitcoin exchange using elliptic curve cryptography. *Future Generation Computer Systems, In Press.*

Wang, S., Zhang, Y., & Zhang, Y. (2018). A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access, (6), pp.38437-38450.*

Wang, Y., & Kogan, A. (2018). Designing confidentiality-preserving Blockchain-based transaction processing systems. International Journal of Accounting Information Systems. *International Journal of Accounting Information Systems, 30, pp.1-18.*

Wang, Y., Shen, Y., Wang, H., Cao, J., & Jiang, X. (2018). MtMR: Ensuring MapReduce Computation Integrity with Merkle Tree-Based Verifications. *IEEE Transactions on Big Data Big Data, 4(3), pp.418-431.*

Weber, R. M. (2018). An Advisor's Introduction to Blockchain. *Journal of Financial Service Professionals, 72(6), pp.49-53.*

Werbach, K. (2018). Trust, but verify: Why the blockchain needs the law. *Berkeley Technology Law Journal, 33(2), pp.487-550.*

Yan, W. (2019). *Introduction to Intelligent Surveillance (3rd Edition)*, Springer.

Yan, Y., & Coffey, A. J. (2014). Audience interactivity on video websites and the business implications for online media platforms. *Journal of Media Business Studies, 11(2), pp.25-56*.

Yuan, Y. & Wang, F. (2018). Blockchain and Cryptocurrencies: Model, Techniques, and Applications. *IEEE Transactions on Systems, Man, and Cybernetics, 48(9), pp.1421-1428.*

Yujian, L., & Bo, L. (2007). A Normalized Levenshtein Distance Metric. *IEEE Transactions on PAMI, 29(6), pp.1091-1095.*

Zhang, J., Wang, Z., Quan, Z., Yin, J., Chen, Y., & Guo, M. (2018). Optimizing power consumption of mobile devices for video streaming over 4G LTE networks. *Peer-to-peer networking and applications, 11(5), pp.1101-1114.*

Zhang, X., & Fan, M. (2018). Blockchain-Based Secure Equipment Diagnosis Mechanism of Smart Grid. *IEEE Access, 6, pp.66165-66177.*

Zhao, S., Wang, B., Li, Y., & Li, Y. (2018). Integrated energy transaction mechanisms based on blockchain technology. *Energies, 11(9).*

Zheng, Q., Li, Y., Chen, P., & Dong, X. (2018). An Innovative IPFS-Based Storage Model for Blockchain. *IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp.704-708.*

Zhou, L., Wang, L., & Sun, Y. (2018). MIStore: a Blockchain-Based Medical Insurance Storage System. *Journal of Medical Systems, 42 (8), pp.17.*

Zhou, L., Wang, L., Sun, Y., & Lv, P. (2018). BeeKeeper: A Blockchain-Based IoT System with Secure Storage and Homomorphic Computation. *IEEE Access, (6), pp.43472-43488.*

Zhou, L., Yan, W., Shu, Y., Yu, J. (2018) CVSS: A Cloud-Based Visual Surveillance System. *International Journal of Digital Crime and Forensics (IJDCF) 10 (1), pp.79-91.*

Zhou, Y., Gu, X., Wu, D., Chen, M., Chan, T. H. & Ho, S. W. (2018). Statistical study of view preferences for online videos with Cross-Platform Information. *IEEE Transactions on Multimedia, 20(6), pp.1512-1524.*