Research On Digital Copyright Management System Based on Blockchain Technology

Xuewang Zhang[1, 2] and Yijun Yin[1]

[1] Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2] Chongqing University, Chongqing 400044, China
zhangxw@cqupt.edu.cn

Abstract. In recent years, the number of digital file has grown explosively, and a large number of digital content is facing copyright registration difficulties, and illegal dissemination. To solve these problems, this paper designed a decentralized digital copyright management system, which based on blockchain technology and IPFS file system. The system utilized the consensus algorithm of blockchain named PBFT, which constructed blockchain system with the lower resource consumption and the bigger throughput. To supervise the DAPP system, we introduce an identification and transaction filter through smart contract which meet the requirement of supervision transaction on blockchain. In order to deal with the copyright information in an efficient, trusted and secure way, strategies of copyright registration and copyright transfer are designed. The results of experiment show that our system can satisfy the requirement of copyright management.

Keywords: Digital Copyright Management, Blockchain, Smart Contract, IPFS.

1 Introduction

With the rapid development of Internet technology, the number of digital content such as e-books, music and video has dramatically increased. According to the fortieth China Internet development statistics report published by CNNIC (China Internet Information Center) in June 2017, internet users are using online literature, online music and online video are up to 35.255 billion, 52.413 billion and 56.482 billion respectively [1]. The internet is the carrier of the digital content, and digital content is easy to be illegally copied and disseminated by others through the network. There are many characteristics such as various infringement methods, difficulty in proof and difficulty in tracking. The traditional offline copyright registration methods have the disadvantages of high cost, multiple procedures and long time-consuming. The online centralized copyright registration system is easily manipulated, without strict traceability. So a large number of digital content copyrights are difficult to protect.

The blockchain technology originated from the paper "Bitcoin: A Peer-to-Peer Electronic Cash System" published by a scholar named Satoshi Nakamoto in 2008 [2], who proposed using proposal P2P network, cryptography, hash algorithm, con-

sensus mechanism, merkle tree and other technologies constructs a distributed ledger. Vitalik Buterin [3] proposed the Ethereum platform in 2014, in addition to the use of the built-in ethcoin, it introduced smart contracts for the blockchain system, making blockchain could load contracts and legal documents into executable programs. Legal affairs will be automatically generated when all conditions are met. The smart contracts have triggered a new thinking about digital economy and digital assets.

For the above advantages of blockchain, many researchers began to explore the application of blockchain in digital rights management. Li [4] proposed a DCI control model of digital works based on blockchain, using smart contracts to implement copyright registration, copyright verification, and copyright transfer without third-party trust, but this model uses POW as a consensus algorithm for the blockchain, which requires large consumption of resources; In 2017 Ebookchain.org launched a digital publishing platform based on blockchain technology and adopted a community incentive way to encourage users to create work and copyright registrations through the platform, but there is no copyright review in the process of copyright registration, which not provide strict copyright proof for creators. Besides, the above ways do not provide regulatory measures to blockchain.

Based on this condition, we completed the following three works: 1. For system architecture design, improved on Ethereum, replacing the POW consensus algorithm with the PBFT consensus algorithm, remove the GAS mechanism, and blocks can store the number of transactions no longer limited by GAS. Base this way build a consortium blockchain to improve system throughput and saves computing resources. Using IPFS as a digital file storage system, a better way to solve file storage; 2. To supervise the blockchain, the role of supervisory agency is introduced, and the identity authentication and transaction filtering strategy are designed through the smart contract. 3. Optimize the copyright management model, and introduce a copyright review agency to audit digital copyright in the system to resolve the digital copyright submitted without review.

2 Related technology introduction

2.1 Data Structure 0f Blockchain

Blockchains divide data into different blocks, each of which is linked to the previous block by specific information [5]. Each block is composed by block header and block body. The block's information includes three categories: the set H related information about block construction, set T related treading information and set U related other information block. The formalized definition of block B is shown in formula (1):

$$B = (B_H, B_T, B_U) \tag{1}$$

The block header contains the previous block's address, timestamp, random number, target hash of the current block, merkle tree root and others [6]. The block body

contains the key information of the transaction and is stored as the merkle tree. The general data structure of the blockchain is shown in figure 1:

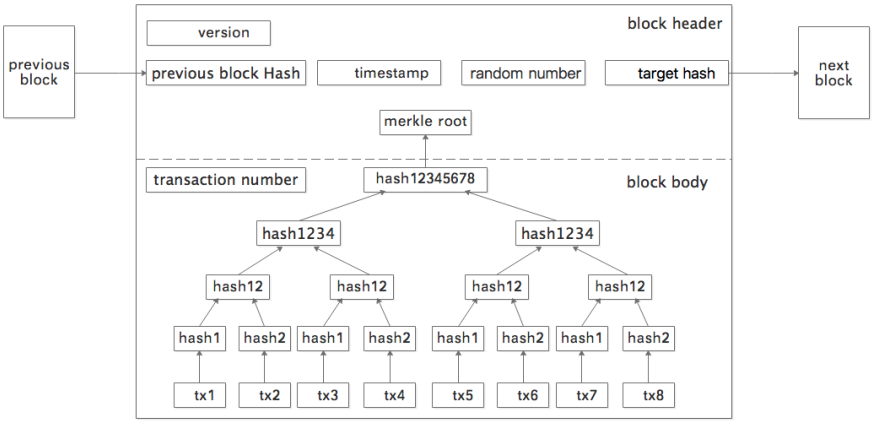

Fig. 1. Data structure of blockchain.

Block store the previous block's information digest in hash, its information digest is stored in header of next block. If you need to change the tread information, you must change all pervious block [7]. At the same time, the structure of merkle tree can quickly verify that if a transaction is in a block. This data structure of blockchain can satisfy the security storage and fast verification of transaction data.

2.2 Data Structure Of Blockchain

The smart contract gives the parties corresponding copyrights and obligations, could manages and controls the execution of transactions in the blockchain system. A Turing complete virtual machine EVM is implemented to execute smart contracts in the Ethereum. The code for the Ethereum contract is written in the low-level language of stack-based bytecode, which is called "Ethereum virtual machine code" or "EVM code". When the Ethereum system executed smart contracts correctly, the correct transaction will change the Ethereum state, and this state also called world state [8,9]. It formal definition as shown in formula (2), S is the state before transaction execution, S' is the state after transaction execution, TX stand for transaction.

$$APPLY(S, TX) \rightarrow S' or\ ERRO \tag{2}$$

Unlike the UTXO model in bitcoin, Ethereum uses Merkle Patricia Tree[4] to maintains the states of accounts and contracts in the blockchain system, which also called world state. The change in states means the modification, addition and deletion of nodes on merkle tree.

2.3 Consensus algorithms

In blockchain system, in addition to network delay and transmission errors, we need to solve the data inconsistency caused by malicious nodes. Therefore, a proper consensus algorithm is needed in blockchain system to ensure that all correct nodes agree on the correct input values.

There are three common consensus algorithms [10,11,12]:

POW(Proof Of Work): Each node competes for writing block rights by calculating a random number that satisfies the rules, and the node on charge broadcast after receiving transaction data. The data will be stored in the node after other nodes have checked it. The public blockchain such as Bitcoin and Ethereum adopt this consensus algorithm.

POS(Proof Of Stack): POS is an upgrade consensus algorithm for POW, through reducing the difficulty of mining according to the time and proportion of each node having tokens, thus speeding up the search for the speed of random numbers. In the future, Ethereum will be ready to switch the consensus algorithm towards POS.

PBFT(Practical Byzantine Fault Tolerance): The protocol contain three phase: pre-preparation phase, preparation phase and confirmation phase. In the pre-preparation phase, the primary node sends pre-prepared messages. The other nodes enter the preparatory phase, after receiving pre-prepared messages. In the preparation phase, the primary node sends message that contain verification record to all nodes. Each node verifies its correctness, and the correct record is saved and sent to the other nodes. Until a node receives 2f different records, each of record is same as the record receive in pre-prepared phase. This node broadcast a confirmation message to others. In the confirmation phase, until each honest node receives 2f+1 confirmation message, the protocol terminates and the nodes agree on the record. IBM's Hyperledger Fabric uses this consensus algorithm.

The specific comparison of the above consensus mechanism is shown in Table 1.

Table 1. Comparison of common consensus algorithms.

Consensus algorithm	Fork	Fault tolerance rate	Number of nodes	Latency	Calculation consumption
POW	hava fork	<50%	more	high	high
POS	hava fork	<50%	more	High	low
PBFT	no fork	<50%	less	low	low

2.4 Consensus algorithms

IPFS is short for Inter Planetary File System, which is a point-to-point, distributed file system [13]. It is different from the HTTP server. When we find a file, look up server IP address, then apply to the server for the file, IPFS find files by looking for the file content, and doesn't care about the file name and path. When we store a file in an IPFS node, it will hash the file and use the hash as the addressing mode for the file. When requests a file through IPFS, it uses a distributed hash table to find the node

where the file resides are. Fetching the file and verifying the data. The IPFS management file can improve the reliability and transmission efficiency when save and distribute file.

3 System design

3.1 System architecture

The system mainly includes two types of nodes: IPFS node and blockchain node. The IPFS node is used to store the digital file, and the blockchain node mainly authenticates the transaction, executes the transaction, writes the transaction into the block, and updates the state.

Three types of role members: (1) general User: submit identity authentication, application for copyright, complete copyright transaction. (2) copyright review agency: review the user's submitted copyright information. (3) supervisory agency: review user submitted identity information and formulate regulatory strategy.

Four smart contracts: (1) Identity authentication contract: mainly stores the user's address, user's identity information and an flag of whether to completed the review; (2) Transaction filtering contract: mainly stores the address of the user's blacklist, and the information about user's invoke authority to the contract method; (3) Copyright recording contract: mainly stores the hash of digital files, basic information of digital file, the address of the owner of the copyright, and the result of the review of the copyright registration; (4) Copyright management contract: The entrance contract of the copyright management service includes the following methods: submit identity authentication, review identity authentication, submit copyright registration, review copyright registration, submit copyright transfer, add filter rule, check whether the user passes the identity verification, and check whether the user is on the blacklist, check the permissions when user invoke contract method, and so on.

The design of the system does not depend on any central server. The overall system architecture is shown in Figure 2.

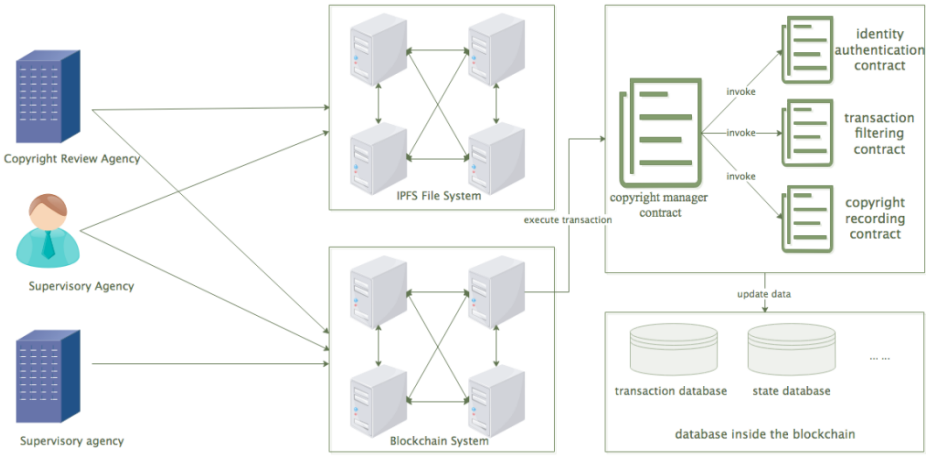

Fig. 2. System Architecture

3.2 User authentication design

In this system, <account address, identity information, review result> is used to identify each user, and the private key is generated by the user in the base on ECDSA-secp256k1 standards and saved by themselves. The corresponding relation between the private key and the user address is shown in formula (3).

$$A(Pr) = B_{96\ldots255}(KEC(ECDSAPUBKEY(Pr))) \tag{3}$$

Pr is the private key obtained by the elliptic curve encryption algorithm ECDSA-secp256k1, ECDSAPUBKEY stand for the method of converting the private key to the public key, KEC stand for the hash processing of the user's public key using keccak as a hash algorithm, $B_{96\ldots255}$ means to take the 96 to 255 bits of the binary number, a total of 160 as the account address, which is a common account address in the blockchain after converting it into a hexadecimal.

The user's authentication process requires the user to submit identity information and the supervisory agency to review it.

1) The user submits identity authentication to the supervisory agency through the blockchain. The specific process is as follows:

$$TX = CREATETX(I_{sender},\ C_{manager},\ FUNC\_SUBMITID,\ Pr_{sender})$$

$$S'_{identity} = APPLY\ (S_{identity}, TX)$$

The user create a transaction and writes a new state to the identity authentication contract $C_{identity}$. The written information includes: identity information I_{sender}, including personal name, ID number, and so on. And finally, the user use own private key Pr_{sender} to sign the transaction, and send the transaction to blockchain system. When the transaction is running, invoke submit identity authentication method *FUNC_SUBMITID* in the copyright management contract $C_{manager}$, and the user's

address and user information are written into the identity authentication contract through the contract internal invoke. TX stand for signed transaction. $S_{identity}$ stand for the state of the identity authentication contract before the transaction is confirmed. $S'_{identity}$ stand for the state after the transaction is confirmed. If the transaction execution successful, a user record will be added in the identity authentication contract: $\{A_{sender}, I_{sender}, Flag\}$. A_{sender} is the address of the user, and the address is obtained by parsing the transaction signature. *Flag* is the state of review.

2) When the user submitted the request, supervisory agency need to review user identity information. The specific process is as follows:

$$TX = CREATETX(result,\ C_{manager},\ FUNC\_REVIEWID,\ Pr_{supervisory})$$

$$R = CHECKSENDER(TX)\ ==\ A_{supervisory})$$

IF R==TRUE

$$S'_{identity} = APPLY\ (S_{identity}, TX)$$

ELSE

ERROR

The supervisory agency reviews the identity information submitted by the user in the blockchain. After the review is completed, the supervisory agency creates a contract transaction to invoke the review identity authentication method *FUNC_REVIEWID* of the copyright management contract, and to write the review result to the identity authentication contract $C_{identity}$. When the transaction has signed by supervisory agency private key $Pr_{supervisory}$, and send to transaction system. The *FUNC_REVIEWID* method will check the address of the sender, and only the transactions sender is supervisory agency's address $A_{identity}$ can successfully invoke this method. When the user authentication is passed, the transaction executed success. The states of identity authentication contract $S_{identity}$ updated to $S'_{identity}$, the *Flag* change to true.

3.3 Transaction supervision design

The supervisory agency adds corresponding filter rules to the transaction filter contract, such as user blacklists and authority of contract method into the transaction filter contract. The contract method associated with the filter rule only allows transaction with the signature of the supervisory agency to invoke. The process of adding filter rules is as follows:

$$TX = CREATETX(I_{rule}, C_{manager}, FUNC\_ADDRULE, Pr_{supervisory})$$

$$R= (CHECKSIGN(TX) == A_{supervisory})$$

IF R ==TRUE

$S'_{filter} = APPLY\ (S_{filter}, TX)$

ELSE

ERROR

The supervisory agency creates a contract transaction to invoke the add filter rule method *FUNC_ADDRULE*. This function writes the new rule I_{rule} into the transaction filter contract through the contract internal invoke and updates user blacklist and user permission table. When user invoke function of copyright registration and transaction, these function will check the transaction sender according to the information in the transaction filtering contract, which decides the transaction whether it is successful or not. so that the supervisory agency can regulate the blockchain.

3.4 Copyright management design

Copyright registration

The user submits the digital file to IPFS file system. A hash is returned as the addressing path after successful submission, then submits the basic information of the digital file and the file hash returned by the IPFS to the blockchain. The specific process is as follows:

FileHash=PUTTOIPFS(File)

$TX = CREATETX(FileHash, I_{digital}\ C_{manager},$
$FUNC\_SUBMITCOPYRIGHT\ , Pr_{sender}\ )$

R1 = FUNC_CHECKUSER(TX)

R2 = FUNC_CHECKAUTHORITY (TX)

IF R1==TRUE && R2==TURE

$S'_{record} = APPLY\ （S_{record}\ , TX）$

ELSE

ERROR

Firstly, the user submits the digital file. PUTTOIPFS stand for submit digital file to IPFS file system, FileHash is returned as the addressing path and the unique identifier for the digital file, $I_{digital}$ is basic information of digital file. After the submission is successful. Then user create a transaction and writes a new state to the Copyright recording contract C_{record}. Finally, use own private key Pr_{sender} to sign the transaction, and send the transaction to blockchain system. The contract transaction invokes the submit copyright registration method FUNC_SUBMITCOPYGITHY of the copyright management contract, during this process will check whether the user passes the identity verification by the FUNC_CHECKUSER method, and check blacklist and

permissions by FUNC_CHECKAUTHORITY. If there are all passed, the transaction will execute success. A new record {FileHash, $I_{digital}$, A_{sender}, Flag} will be added to the copyright record contract, A_{sender} is address of user. If any of check not satisfied during the process, the transaction will fail and the state of the contract will not be update.

Copyright review

After the user submits the application, the copyright review digital file and submit information. The specific process of digital copyright review is as follows:

$TX = CREATETX(result, C_{manager},\ FUNC\_REVIEWCOPYRIGHT,\ Pr_{review})$

$R = (CHECKSENDER(TX) == A_{authority})$

IF R ==TRUE

$S'_{record} = APPLY(S_{record}, TX)$

ELSE

ERROR

Copyright transfer

After consultation, the copyright owner creates the transaction and transfers the copyright to others The specific process is as follows:

$TX = CREATETX(FileHash, A_{recipient}, C_{manager}, FUNC\_TRANSFER, Pr_{sender})$

R1 = FUNC_CHECKUSER(TX)

R2 = FUNC_CHECKAUTHORITY(TX)

R3 = (FUNC_CHECKOWNER(FileHash)== A_{sender})

IF R1==TRUE && R2==TURE&&R3==TRUE

$S'_{record} = APPLY\ (S_{record}, TX)$

ELSE

ERROR

The copyright owner creates a contract transaction, which changes the copyright owner address to the recipient's address $A_{\text{recipient}}$, signatures the transaction with the private key of the copyright owner Pr_{sender} , submits the transaction to the blockchain system, completes the transfer of the copyright and updates the state of contract. In the execution of the method, the transaction sender will be verified to check whether the user has the copyright of the digital by *FUNC_CHECKOWNER*, and when

FUNC_CHECKUSER, *FUNC_CHECKAUTHORITY*, *FUNC_CHECKOWNER* are all passed, the transfer of copyright will be successful.

4 Experiments and analysis

4.1 Calculation resource consumption analysis

The experimental environment is Intel i7 7700 processor, 16GB memory. Through the docker tool, we build 4 nodes separately, and two different blockchain systems. One system is to use POW as a consensus algorithm, like Ethereum, and the other system use PBFT as a consensus algorithm. When the two groups of blockchains system are initialized，The CPU utilization percentage of both are shown in Figure 3.

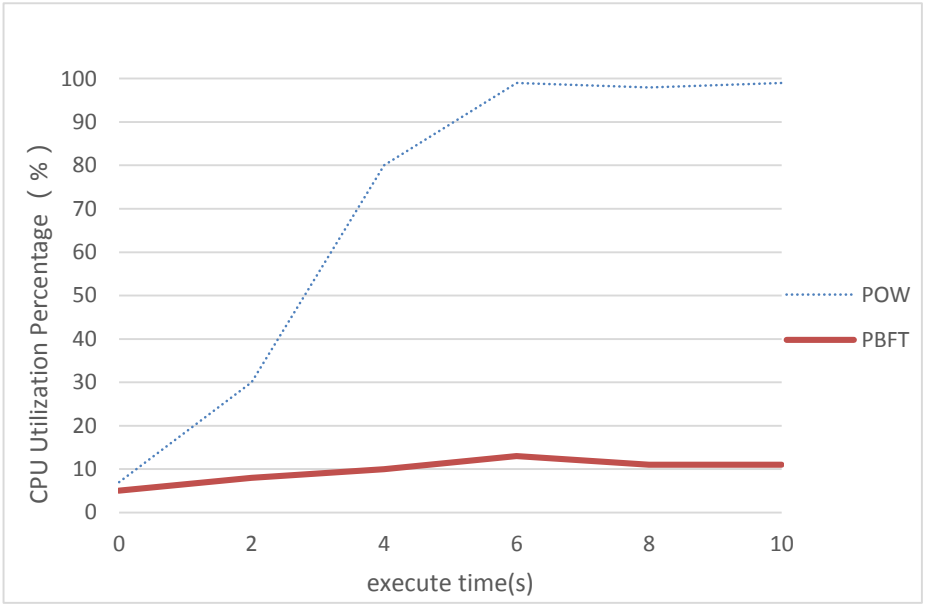

Fig. 3. Comparison of CPU utilization.

As shown in the above figure, the CPU utilization percentage of the blockchain system which using POW as a consensus algorithm rapidly increased after the mining is started, which is close to 100%. The blockchain system which using PBFT as a consensus algorithm does not need to complete the calculation of random numbers, and is mainly responsible for communication between nodes and EVM execution. The CPU usage rate is stable at about 10% - 12%. The CPU utilization of blockchain system using PBFT as a consensus algorithm is lower, which greatly reduces the resource consumption for achieving consensus.

4.2 Throughput analysis

This test we use the docker tool to build 4 nodes to satisfy the fault tolerance requirement of 3f+1 in the PBFT algorithm, and set the block output time to 10s.The test method is as follows: Ni is the number of transactions that have been sent in unit time, and as time goes on, Ni continues to increase until Ni is slightly larger than the number of transactions in the block and the number of transactions in each block tends to a stable range. The results of the experiment are shown in Figure 4.

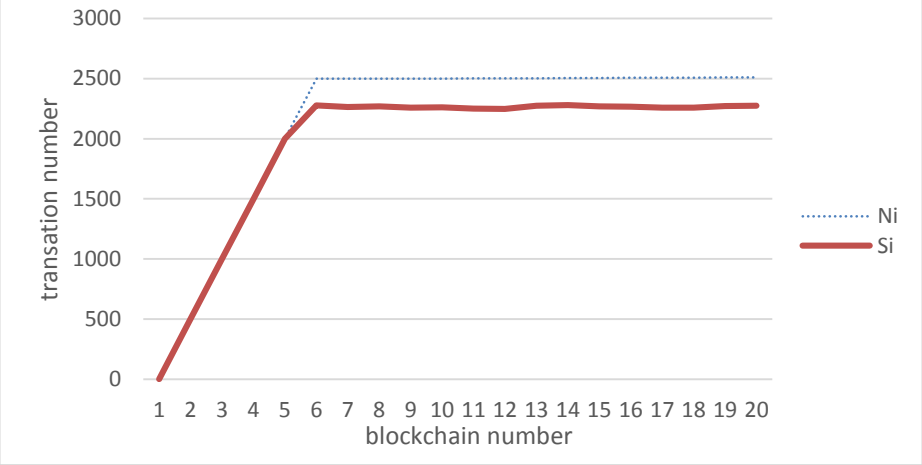

Fig. 4. System throughput.

According to Figure 3, TPS is calculated according to the formula (4) for the range data that is stabilized, and the block-out time for the system has been set to 10s, so in the formula (4), t_i can be replaced with 10s for calculation.

$$\mathrm{TPS} = \sum_{i=1}^{n} t_i / \sum_{i=1}^{n} S_i \tag{4}$$

After calculation, about 220 TPS can be achieved in the experimental environment, enough to satisfy the system requirements of copyright management. PBFT as a consensus algorithm, its transaction speed is greatly affected by the network environment, and different transactions in different service scenarios also will affect performance, compared with Ethereum use POW consensus algorithm and GAS mechanism to limit the number of transactions in the block, the throughput of the system is greatly improved.

4.3 Security Analysis

The digital copyright management system architecture consisted of an IPFS file system and blockchain system, both of which use a decentralized, distributed architecture that can effectively defend against DDoS attacks. The PBFT protocol requires that in

a distributed system with 3f+1 nodes, the malicious nodes do not exceed the f. At the confirmation stage, each honest node receives 2f+1 confirmation messages, and each node can agree on the written data. Therefore, as long as the malicious nodes in the system are less than f, the normal operation of the system can be correct execution. And the data information in the blockchain is encrypted by private key, which ensures the confidentiality of the data in the blockchain.

5 Conclusion

In this paper, we utilize blockchain technology and IPFS file system to design a decentralized copyright management system, using the IPFS file system to solve the storage of massive digital files. As the blockchain system is difficult to supervise, we introduced identity authentication and transaction filtering strategy in the contract layer. The whole copyright management process is carried out on the blockchain, with high credibility and safety. Finally, the experimental analysis shows that the performance and security of the system can satisfy copyright management, and consumes less computing resources.

Acknowledgments. .This work was partly financially supported through grants from the National Natural Science Foundation of China (No. 61571072), Major Science and Technology Project for "Innovation of Common Technology in Key Industries" of Chongqing (No.cstc2017zdcy-zdzxX0013), and Chongqing Research Program of Basic Research and Frontier Technology (No. cstc2013jcyjjA40012).

References

1. China Internet Network Information Center, http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201708/P020170807351923262153.pdf, last accessed (2017)
2. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf (2008)
3. Buterin, V.: A next-generation smart contract and decentralized application platform, https://ethereum.github.io/yellowpaper/paper.pdf (2014)
4. Li, Y., Huang, J.K., Wang, R., Deng, E.Y.: DCI control model of digital works based on blockchain. Journal of Computer Application 37(11), 2381-2387(2017)
5. Cai, W.D., Yu, L., Leng, K.J., Wang, C.F.: Blockchain Application Development Techniques. Journal of Software 45(2), 40-47(2018)
6. Yuan, Y., Wang. F.Y.: Blockchain: The State of the Art and Future Trends. Acta Aotomatica Sinica 42(4), 481-494(2016)
7. Bi, Y., Zhou, B., Leng, K.J, Wang, C.F.: Public Blockchain of Pharmaceutical Business Resoueces Based on Double-chain Archtecture. Computer Science 45(2), 40-47(2018)
8. Lu, J., Song, B., Zhou, Z.M.: Smart Contract for Electricity Transaction and Charge Settlement Based on Blockchain. Computer Systems & Applications 26(12), 43-50(2017)
9. Shao, Q.F., Jin, C.Q., Zhao, Z., Qian, W.N., Zhou A.Y.: Blockchain: Architecture and Research Progress. Computer Systems & Applications 26(4), 1-8(2017)

10. Huang, Q.B., An, Q.W., Su, H.L.: Study and Realization of an Improved PBFT Algorithm as an Ethereum Consensus Mechanism. Computer Applications and software 34(10), 288-293+287 (2017)
11. Han, X., Liu, Y.M.: Research on the Consensus Mechanisms of Blockchain Technology. Netinfo Security 201(9), 147-152 (2017)
12. Xia, Q., Zhang, F.J., Zuo, C.: Review for Consensus Mechanism of Cryptocurrency System. Chinese Journal of Computer 26(04) 1-20(2017)
13. Benet, J.: IPFS - Content Addressed, Versioned, P2P File System. https://arxiv.org/pdf/1407.3561.pdf. (2014)