Received July 13, 2020, accepted August 28, 2020, date of publication September 1, 2020, date of current version September 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020836

A Secure Storage Strategy for Blockchain Based on MCMC Algorithm

PENG ZHAO[1], HONGBING CHENG[1,2], YICHENG FANG[1], AND XIAOQING WANG[1]
[1]College of Computer Science, Zhejiang University of Technology, Hangzhou 310037, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Corresponding author: Hongbing Cheng (chenghb@zjut.edu.cn)

This work was supported in part by the Nature Science Foundation of Zhejiang Province under Grant LY19F020023; in part by the National Nature Science Foundation of China under Grant 61402413, Grant 61706342, Grant 61772026, and Grant 61603340; and in part by the Joint Funds of the National Natural Science Foundation of China under Grant U1509214.

ABSTRACT The exponential growth of storage space in blockchain network has become a serious problem to hinder the distribution of blockchain and the expansion of blockchain nodes. In this paper. We propose a security strategy for distributed storage blockchains, which can delete part of blockchains so that nodes only store part of a blockchain. We design a kind of semi-full node between full node and light node according to the requirement of the strategy, besides describe the process of deleting block and synchronizing block, and the running logic of the semi-full node. Finally, we perform comprehensive experiments of the truncated MCMC random algorithm. The results show that in the case of multi-node, the truncated block will not affect the block chain network. Compared with the traditional block design, our storage strategies can reduce storage requirements under most of situation, thus enable blockchains to be deployed on mobile or smaller storage computers.

INDEX TERMS Blockchain storage, distributed storage, MCMC algorithm.

I. INTRODUCTION

With the rapid development of blockchain technology, the scale of various blockchain projects is gradually increasing, and the demand for computer storage is also increasing. One of the most important technologies in blockchain is tamper-proofing; in short, it is for each node in blockchain network to store the same blockchain data file. This distributed storage methods ensure that the blockchain data can not be easily changed, and the more nodes store data, the more difficult to tamper with the data. However, the storage capacity of blockchain also becomes a serious problem which hinders the development of blockchain. The original blockchain project, Bitcoins [1], has approached 300 GB [2] of blockchain capacity in the past 11 years, while another well-known project, Ethereum [3], has approached 200 GB [4] of blockchain capacity as well. Figure 1 shows the storage and memory requirement of Bitcoins and Ethereum will continue to grow. However, the memory capacity of civilian computers, which have increased over the past decade, will not be able to keep pace with the rapid growth of

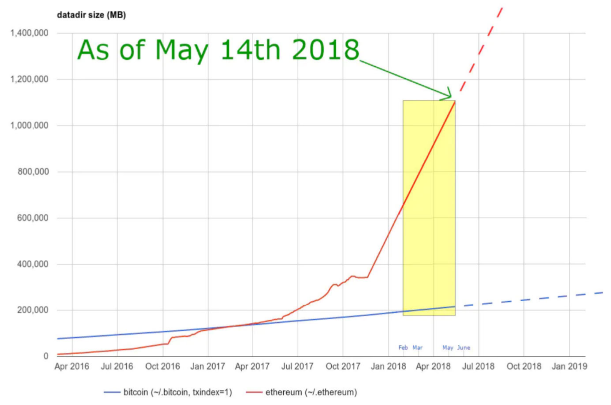


FIGURE 1. **Storage and memory growth for Bitcoin and ethereum.**

the blockchain. one 2019 Lenovo IdeaPad S145 15." laptop computer costs about $376 on Amazon has reached almost 4GB DDR4 RAM, 500GB HDD. To become a full node in Bitcoin or Ethereum, we need buy a 500 GB laptop, and if we want to continue to be a blockchain node then we need to keep increasing the storage capacity. General-purpose computer equipment simply does not have enough space to store the ever-increasing blockchain data if the computer system

The associate editor coordinating the review of this manuscript and approving it for publication was Shancang Li.

has processed these blockchain applications, this computer device becomes a dedicated block chain processing equipment, which undoubtedly increases the cost. So how to reduce the memory capacity in the block chain node is a very important problem in the development of the block chain.

The information stored in the blocks of Bitcoin and other blockchains can be described as transaction information, with a storage size of only a few hundred bytes [2]. As more and more nodes are deployed on the blockchain, the competition among the nodes is increasing, and some nodes will become lightweight nodes [5] or join the mining pool [6] due to lacking of computing capacity or storage space.

In this paper, we design a new blockchain storage strategy, which can reduce the block information in different nodes according to the random algorithm. According to the working mechanism of the strategy, we design a semi-full node different with the lightweight node and the full node. The state of the semi-full node is the full node state at the beginning of its establishment, and it will download the complete blockchain information for synchronization. However, after a certain amount of blocks are synchronized, they will be converted to semi-full node state, and then the blocks will be verified and synchronized before its random deletion. At the same time, the node will record the deleted block index, and the node will get the incomplete block blockchain. Once the synchronization is successful (downloaded to the latest block), then the semi-full node will delete the block that has not been randomly processed before updating a certain block. The semi-full node is different with the lightweight node. The semi-full node has the ability to query transaction information at the same time, so it can participate in the verification and mining game of the blockchain system (because the semi-full node stores certain block information of the blockchain, when querying the reliability of a transaction, it will first query the transaction from its own block, and then visit if the query fails Ask the transaction data on the blockchain of other nodes, return and verify the data and return the final result. Although it reduces the success rate of mining game in a certain probability, it retains the possibility of winning. We construct the strategy to focus on the reduction of the size of the blockchain storage, so that more people will be able to participate in the establishment of the blockchain network. In blockchain system, in order to ensure the security of the whole network, the number of honest nodes must reach a certain scale and the computing capability of these honest nodes is not less than 51% of the whole network. However, considering the block chain storage problem, more nodes will tend to become lightweight nodes, because the current mainstream block chain public chain projects are too concentrated on computing capability, a few large node organizations almost control the capability of the entire blockchain network, it means that they hold the blockchain's rewards, which reduces the incentive to become a full node. So we only need to ensure that each node has at least part of the reliable block information, through the number of nodes advantage to make up for the lack of a single node storage. Compared with the block chain fragmentation system which needs to protect most nodes, we can randomly allocate less block information to one node to ensure the security of the whole network under the condition of less security situation. The primary contribution of the paper is to propose a new node model semi-full node on the basis of existing blockchain node function. And put forword the conversion model of full node, light node and semi-full node, not only can greatly reduce the size of blockchain storage capacity, but also fair deletion of block information based on MCMC random algorithm. Finally more nodes can independently participate in the mining game.

In the following chapters, we will introduce the blockchain storage strategy and the working mechanism of semi-full nodes in detail. We will also show the data requirements compared with the traditional blockchain (Bitcoin). In Section 2, we introduce the current research on blockchain storage and consensus nodes. In Section 3, the theoretical basis of the stochastic algorithm is given. In Section 4, the storage strategy method and algorithm are described. In Section 5, the storage strategy is tested and the results are analyzed. Section 6 summarizes the whole paper and prospects the future research.

II. RELEVANT RESEARCH

Recently, many scholars have accomplished some research on improving the performance of blockchain, such as weighted model [7], [8], blockchain segmentation [9], [10], consensus mechanism improvement [11]–[13], etc. All these methods try to reduce the burden of a single node, to ensure the low storage of the blockchain system and to improve the node performance at the same time.

In the weighted node blockchain model, different weights of nodes lead to different chances of winning in the mining game. The higher the weight, the greater probability it will win in the game [1]. In lightweight node system, [5] is a good example of weighting model. In their blockchain model, the model's lightweight nodes (lightweight nodes) do not need to store any blockchain information blocks. When lightweight nodes need to verify new transactions, they mainly rely on simple payment verification (SPV) to query the reliability of transactions [14]. Regardless of the total block size of the blockchain, a lightweight node only needs to store 4 megabytes of blockchain information every year (this part of information is the block header information of the blockchain) [14], but the lightweight node cannot verify the new block, only can passively synchronize the blocks of other nodes, and may be cheated by the nodes with complete blocks. In the blockchain combined with the dpos consensus mechanism [15], the blockchain system selects a fixed number of representatives through the shareholders. These representatives will generate blocks in turn according to a certain order. Of course, these representatives are not immutable. If their performance in the blockchain network can no longer be trusted by the shareholders, the authority of the representatives will be cancelled. Dpos has good performance, because the representative nodes usually have computers with high computing capability, sufficient storage space and high network bandwidth. Therefore, the security of blockchain network depends on these representative computers to a large extent. In this way, the control of the network is too centralized, which deviates from the idea of decentralization.

For blockchain segmentation, it is to divide the blockchain into multiple segments according to the dynamic order of blockchain blocks. In Xu and Huang [10] paper, when a node joins the blockchain network and provides storage evidence to the network every time, it will use PoW (proof of work), which can effectively avoid Sybil attack [16], and also ensure the blockchain attackers will not account for 51% of the total blockchain network computing capability. Xu also made a research on blockchain segmentation [17] and used the hypothesis of marking different nodes of blockchain segmentation. In paper [10], it also divides the participating nodes into different classes and ensures that each part of the blockchain is stored in a node of each class. The node will retain the blockchain fragment in the edited mining game and get rewards.

As for the consensus mechanism, PoW and PoS are common consensus mechanisms in public chain. In order to solve the shortcomings of PoW and PoS algorithms, Kim *et al.* proposed a method of proof of probability (PoP, Proof-of-Probability) in [11], This method uses the hash sorting algorithm to sort the encrypted hash in each node, creates a block for the first node to decrypt the actual hash, and sets the waiting time when decrypting the next hash to limit the excessive computing capability competition. In addition, when more network participants have interests, the probability of acquiring crypto currency will become higher. Kejiao Li *et al.* Proposed a consensus algorithm (PoV, proof of vote) for consortium blockchain. POV divides the participants into different security identities based on voting activities and voting mechanism [12]. The submission and verification of blocks are decided by the voting of the consortium organization in the consortium, which not only ensures the fairness within the consortium, but also promotes the development of the consortium. Hubert Ritzdorf proposed comrade in [13] by using blockchain technology, reached a consensus on access control decision-making in cloud storage platform and converted it into storage access control rules.

III. PRELIMINARY

A. RANDOM ALGORITHMS

MCMC is a method of sampling by constructing a suitable Markov chain and then using Monte Carlo method for integration calculation. It is known that the Markov Chain can converge to a stationary distribution [18], [19], so we establish a π as stationary. The distributed Markov Chain can reach a stable state after running the chain for enough time. The value of the Markov Chain is equivalent to taking samples in the distribution $\pi(x)$. Therefore, MCMC is a random simulation method suitable for Markov Chains.

The first MC: Monte Carlo [20]. let us use random sampling to solve the calculation problem. In MCMC, it means that the posterior distribution is used as a random sample generator. We use it to generate samples, and then use these samples to estimate some interesting calculation problems (features, predictions).

The second MC: Markov Chain [21]. The second MC is the key to this method, we can find that in the first MC we need to use the posterior distribution to generate random samples. When these samples are independent, the posterior distribution is too complex. Therefore, we use the law of large numbers for sampling, and the sample mean will converge to the expected value. If the samples are not independent, then we need to use Markov Chain to sample, and use Markov Chain's stationary distribution to sample the complex posterior distribution. The most commonly used algorithms are metropolis Hastings (M-H) algorithm [22] and Gibbs sampler [23].

The definition of Markov Chain is as follows:

Let $\theta(x)$ be a random process if it satisfies the following properties:

$$\begin{aligned} p(\theta(t+h) &= xt+h|\theta(s) = xs, s \leqslant t) \\ &= p(\theta(t+h) = xt+h|\theta(t) = xt) \quad \forall h > 0 \end{aligned} \tag{1}$$

Sampling definition based on Markov Chain

If we get the Markov Chain state transition matrix corresponding to a stationary distribution, it is easy to adopt this stationary distribution sample set. The definition of stationary distribution is as follows:

If the aperiodic Markov state transition matrix and probability distribution are satisfied for all:

$$\pi(m)Z(m,n) = \pi(n)Z(n,m) \tag{2}$$

The probability distribution is said to be a steady distribution of the state transition matrix.

From the meticulous stationary conditions, the proof is as follows:

$$\begin{aligned} \sum_{m=1}^{\infty} \pi(m)Z(m,n) = \sum_{n=1}^{\infty} \pi(n)Z(n,m) = \pi(n) \sum_{n=1}^{\infty} Z(n,m) \\ = \pi(n) \end{aligned} \tag{3}$$

That is, the convergence properties of Markov Chains are satisfied. It can also be said that Markov Chain sampling is a process of finding a stable distribution and then finding the state transition matrix of the Markov Chain.

B. ENCRYPTION ALGORITHM

Elliptic curve digital signature algorithm (ECDSA) bases on elliptic curve cryptography (ECC). It consists of four algorithms: parameter generation, key generation, signature generation and signature verification [24], which has the advantages of fast speed, high strength and short size signature. For example, Bitcoin uses ECDSA based on the curve secp256k1 [25]. The elliptic curve digital signature algorithm and steps are as follows:

Curve formula:

$$y^2 = (x^3 + a * x + b) \mod p \tag{4}$$

Setup: parameter generation. Output common parameters, where a and b are the parameter coefficients of the ellipse, p is the selected finite field (primary number less than 160 bits (binary)), G is the base point (starting point or abscissa), and n is the order of G.

KeyGen (d, pp): Key generation algorithm. Randomly choose integer calculation, where pp is the public key and d is the private key. That is, input parameter ty and random integer d, output public and private key pair.

Sig (d, m): signature algorithm. Enter the private key d and the message m to be signed, and output the signature S for the message m.

Verify (Q, S): verification algorithm. Enter public key Q and the signature S of message m. If the signature is legal, the output is 1; otherwise it is 0.

TABLE 1. Models of consensus.

Algorithm	PoS	DPoS	Casper	PoET	PBFT
Decentralized	complete	complete	complete	semi	semi
Tokens	yes	yes	yes	no	no
Evil number	51%	51%	51%	51%	33%
Performance	relatively high	high	relatively high	high	high
Technical maturity	mature	mature	not applied	not applied	mature

C. CONSENSUS MECHANISM

The distributed storage strategy we designed is applicable to a variety of consortium blockchain and public blockchain models, as shown in Table 1, such as Proof of Work, Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Casper, Proof of Runtime (PoET) and PBFT [15]. Generally, each is divided into three parts: verifying identity, selecting different types of nodes, and synchronizing data in the blockchain.

When the blockchain adopts the consortium blockchain model, most of its consortium members are credible and effective, such as governments, service operators, and large enterprises. In this case, the design of the storage method of the deleted node block is lower than in the past, and there is no need to deliberately control the number of deleted blocks. When the blockchain adopts a public blockchain, it is necessary to consider the possibility of a large number of attackers in the blockchain network, so we strictly control the deletion of blocks to avoid excessive reduction due to cyber security. In addition, in order to reduce the storage and calculation burden of the blockchain, data generated by multiple terminals is analyzed and recorded by the edge node. In this way, the consensus algorithm is only used to verify identity and store authentication logs in the blockchain to achieve data traceability and prevent data tampering. This paper mainly describes the verification, synchronization and deletion of blocks under the public blockchain.

D. MERKLE TREE

The Merkle tree [26] proposed by Ralph Merkle was originally used to generate a summary of the digital certificate directory. Follow-up researchers proposed many improvements, such as the simplest binary Merkle tree Bitcoin. Each node in the tree is a hash value, and each leaf node corresponds to the SHA256 hash value of the transaction data in the block; after the values of the two child nodes are connected, the parent node can be obtained by hash operation The value of the point; in this way, two or two hash operations are repeatedly performed until the root hash value is generated, which is the Merkle root of the transaction. Through the Merkle root, any tampering of transaction data within the block will be detected, thereby ensuring the integrity of the transaction data. This operation does not require other nodes on the tree to participate, and only judges based on the branch path from the transaction node to the Merkle root path, and can confirm whether a transaction exists in the block based on simple payment verification. The operation of the Merkle tree is shown in Figure 2.

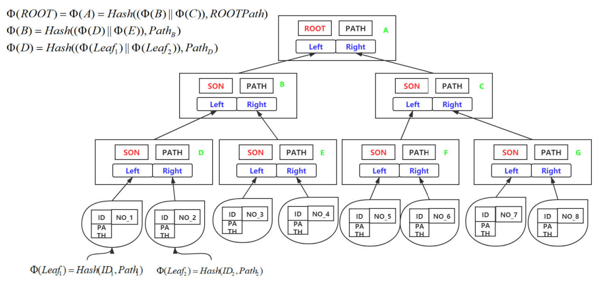

$\Phi(ROOT) = \Phi(A) = Hash((\Phi(B) \| \Phi(C)), ROOTPath)$
$\Phi(B) = Hash((\Phi(D) \| \Phi(E)), Path_B)$
$\Phi(D) = Hash((\Phi(Leaf_1) \| \Phi(Leaf_2)), Path_D)$
ROOT
PATH
A
SON
PATH
B
C
D
E
F
G
Left
Right
ID
NO_1
NO_2
NO_3
NO_4
NO_5
NO_6
NO_7
NO_8
PA TH
$\Phi(Leaf_1) = Hash(ID_1, Path_1)$
$\Phi(Leaf_2) = Hash(ID_2, Path_2)$

FIGURE 2. Merkle Tree data structure diagram.

The value of the leaf node:

$\Phi(Leaf_i) = Hash(ID_i, Path_i), i = 1, 2 \cdots, n$

The values of the son nodes (or intermediate nodes) are:

$\Phi(M) = Hash(\Phi(M_{left})||\Phi(M_{Right}), Path_M)$

The value of the root node is:

$\Phi(ROOT) = Hash(\Phi(ROOT_{left})||\Phi(ROOT_{Right}), ROOTPath)$

As shown in the Figure 2, the Merkel tree is built when $n = 8$:

The value of the leaf nodes:

$\Phi(Leaf_i) = Hash(ID_i, Path_i), i = 1, 2 \cdots, 8$

The value of son node D is:

$\Phi(D) = Hash(\Phi(Leaf_1)||\Phi(Leaf_2), PathD)$

The value of son node B is:

$\Phi(B) = Hash(\Phi(D)||\Phi(E), PathB)$

The value of the root node is:

$\Phi(ROOT) = \Phi(A) = Hash(\Phi(B)||\Phi(C), ROOTPath)$

MT is a kind of tree, most of which are binary trees or multi-trees. No matter what kind of tree trees, it has all the characteristics of the tree structure; The value of the leaf nodes on Merkle Tree is the unit data or unit data HASH in the data set.

The value of nonleaf node is calculated according to the hash value of all leaf nodes connected by the node.

In some specific applications, if extra security requirements is needed, we can use more reliable hash encryption methods, such as SHA-2 and MD5. But if the actual demand is only to prevent data from being damaged or tampered with, some checksum algorithms with low security but high efficiency can be used instead [27], such as CRC.

IV. STRATEGY MODEL

The distributed storage strategy will divide some blockchain into segments. The size and number of blockchain segments are dynamically adjusted according to the number and occupancy of nodes in the blockchain. Each node randomly stores one or more blockchain segments, and also stores the block header of each block in the main chain.

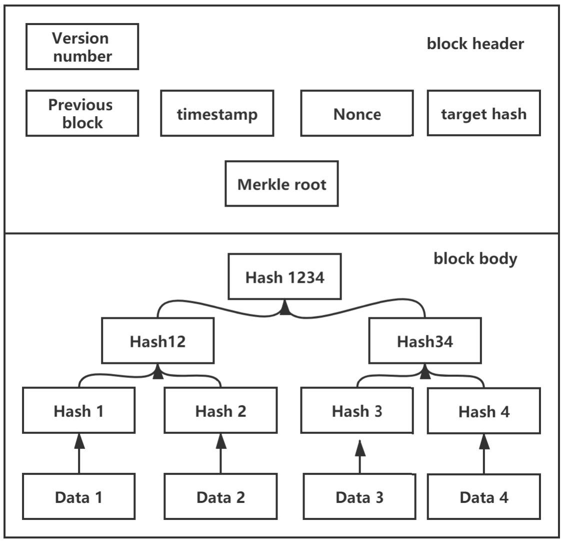


FIGURE 3. Block header of the blockchain.

A. BLOCK HEADER

The Simplified block header is shown in Fig.3. Each block header includes:

1. version number: indicates which set of block validation rules to follow.

2. previous block: a 256-bit hash value that points to the previous block. This item is necessary to ensure the chain structure of the blockchain and is also a prerequisite for the uniqueness of the blockchain data.

3. timetamp: the approximate creation time of the block.

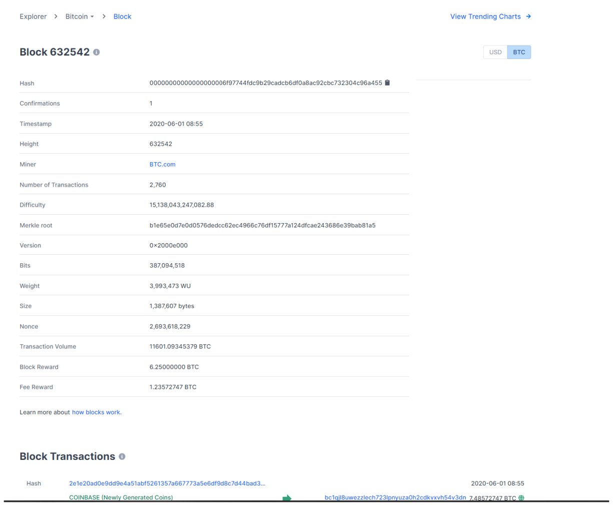
4. Nonce: random number in the block header.

5. target hash: target threshold of a valid block hash.

6. Merkle root: the root hash value of the blockchain Merkle tree. This root stores the generated Merkel tree root hash transaction list for transaction query.

Figure 4 presents a basic structure of the Bitcoin block header. The information about the actual block of Bitcoin can be found on https://www.blockchain.com/btc/ as shown in Figure 5.

In Figure 4 and 5, it can be found that in a Bitcoin block, the block are composed the header information (Figure 3) and the transaction information. The transaction data (Block Transactions) as shown in Figure 5 is data1, data2, which are in Figure 3.



Explorer › Bitcoin ▾ › Block   View Trending Charts →

Block 632542   USD BTC

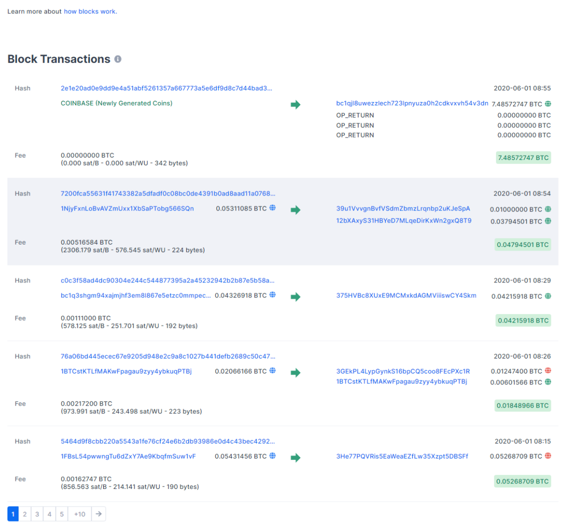
Hash	0000000000000000006f97744fdc9b29cadcb6df0a8ac92cbc732304c96a455
Confirmations	1
Timestamp	2020-06-01 08:55
Height	632542
Miner	BTC.com
Number of Transactions	2,760
Difficulty	15,138,043,247,082.88
Merkle root	b1e65e0d7e0d0576dedcc62ec4966c76df15777a124dfcae243686e39bab81a5
Version	0×2000e000
Bits	387,094,518
Weight	3,993,473 WU
Size	1,387,607 bytes
Nonce	2,693,618,229
Transaction Volume	11601.09345379 BTC
Block Reward	6.25000000 BTC
Fee Reward	1.23572747 BTC

Learn more about how blocks work.

Block Transactions

```
Hash   2e1e20ad0e9dd9e4a51abf5261357a667773a5e6df9d8c7d44bad3...                        2020-06-01 08:55
       COINBASE (Newly Generated Coins)          bc1qjl8uwezzlech723lpnyuza0h2cdkvxvh54v3dn 7.48572747 BTC
```

FIGURE 4. Bitcoin block.



Learn more about how blocks work.

Block Transactions

```
Hash   2e1e20ad0e9dd9e4a51abf5261357a667773a5e6df9d8c7d44bad3...                               2020-06-01 08:55
       COINBASE (Newly Generated Coins)                   bc1qjl8uwezzlech723lpnyuza0h2cdkvxvh54v3dn 7.48572747 BTC
                                                          OP_RETURN                                  0.00000000 BTC
                                                          OP_RETURN                                  0.00000000 BTC
                                                          OP_RETURN                                  0.00000000 BTC
Fee    0.00000000 BTC                                                                                7.48572747 BTC
       (0.000 sat/B - 0.000 sat/WU - 342 bytes)

Hash   7200fca55631f41743382a5dfadf0c08bc0de4391b0ad8aad11a0768...                               2020-06-01 08:54
       1NjyFxnLoBvAVZmUxx1XbSaPTobg566SQn   0.05311085 BTC   39u1VvvgnBvfVSdmZbmzLrqnbp2uKJeSpA    0.01000000 BTC
                                                             12bXAxyS31HBYeD7MLqeDirKxWn2gxQ8T9    0.03794501 BTC
Fee    0.00516584 BTC                                                                                0.04794501 BTC
       (2306.179 sat/B - 576.545 sat/WU - 224 bytes)

Hash   c0c3f58ad4dc90304e244c544877395a2a45232942b2b87e5b58a...                                  2020-06-01 08:29
       bc1q3shgm94xajmjhf3em8l867e5etzc0mmpec... 0.04326918 BTC   375HVBc8XUxE9MCMxkdAGMViiiswCY4Skm 0.04215918 BTC
Fee    0.00111000 BTC                                                                                0.04215918 BTC
       (578.125 sat/B - 251.701 sat/WU - 192 bytes)

Hash   76a06bd445ecec67e9205d948e2c9a8c1027b441defb2689c50c47...                                 2020-06-01 08:26
       1BTCstKTLfMAKwFpagau9zyy4ybkuqPTBj   0.02066166 BTC   3GEkPL4LypGynkS16bpCQ5coo8FEcPXc1R    0.01247400 BTC
                                                             1BTCstKTLfMAKwFpagau9zyy4ybkuqPTBj    0.00601566 BTC
Fee    0.00217200 BTC                                                                                0.01848966 BTC
       (973.991 sat/B - 243.498 sat/WU - 223 bytes)

Hash   5464d9f8cbb220a5543a1fe76cf24e6b2db93986e0d4c43bec4292...                                 2020-06-01 08:15
       1FBsL54pwwngTu6dZxY7Ae9KbqfmSuw1vF   0.05431456 BTC   3He77PQVRis5EaWeaEZfLw35Xzpt5DBSFf    0.05268709 BTC
Fee    0.00162747 BTC                                                                                0.05268709 BTC
       (856.563 sat/B - 214.141 sat/WU - 190 bytes)

1 2 3 4 5 +10 →
```

FIGURE 5. The block transaction section on Bitcoin.

Block transactions occupy the largest storage space in a block. In general, it consumes more than 90% of the entire block storage space. Therefore, the purpose of our strategy is to randomly delete block transactions data and only store the header information of block.

B. SEMI-FULL NODES FOR STRATEGY

To ensure the versatility of the strategies we design, we add a new type node suitable for our strategy without changing the functions of existing nodes.

1) NODE CATEGORY

a: FULL NODE

A node with a complete blockchain ledger. The full node synchronizes all blockchain data. It independently verifies all transactions on the blockchain and updates data inreal time, and is mainly responsible for the broadcasting and verification of blockchain transactions.

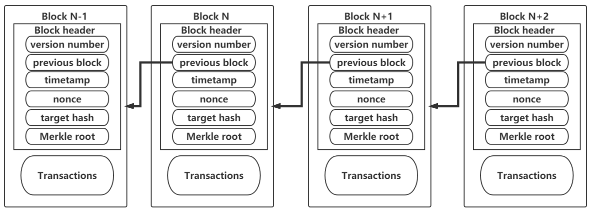


FIGURE 6. Blockchain of undeleted blocks.

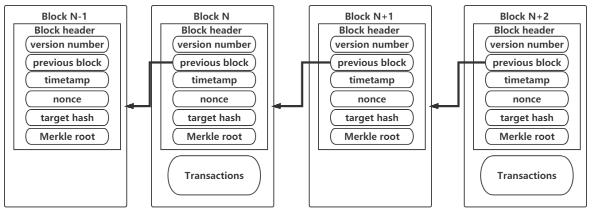


FIGURE 7. Blockchain with deleted blocks.

b: LIGHTWEIGHT NODE (LIGHT NODE)

Light nodes only synchronize block header information and do not need to synchronize transaction information in the block. It takes up a small amount of memory and is suitable for use on mobile devices. But light nodes cannot independently broadcast and verify blockchain transactions. This paper defines a new node type, semi-full nodes.

c: SEMI-FULL NODES

Semi-full
nodes only need to synchronize and verify part of the data on the blockchain, and can be responsible for broadcasting and verifying blockchain transactions. It is suitable for deployment on some mobile devices and low-capacity computers.

2) THREE DIFFERENT CASES OF CONVERSION TO SEMI-FULL NODES

a: FULL NODE CONVERT INTO SEMI-FULL NODE

The conversion of a full node to a semi-full node is relatively complicated. It can be converted only when the blockchain network conforms the security condition:

$$p_1(AN-1)+p_2(HAN+1)\geq k \tag{5}$$

When the conditions are satisfied, the block deletion of the node starts. Figure 6 and Figure 7 shows block deletion flow charts. In Figure 7, some block transaction storage parts in Figure 6 have been deleted.

Specific deletion process:

(1) According to the current difficulty (target hash) and that computing capability of the entire blockchain network, calculate the approximate maximum allowed block deletion probability q and the non-deleted block value BND (Not randomly to delete the latest block). This can improve query efficiency, because the latest transaction tends to refer to the most recent transaction, which is shown in Figure 8.

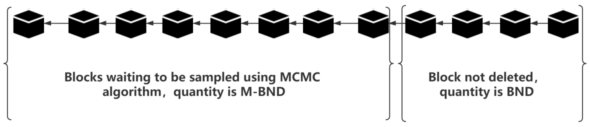

Blocks waiting to be sampled using MCMC algorithm, quantity is M-BND
Block not deleted, quantity is BND

FIGURE 8. Node storage block at the beginning of the deletion from node type to semi-full node.

(2) All nodes sample $M - BND$ (M is the total number of blocks, the current blockchain height) blocks after the creation block using MCMC algorithm substituted with q probability. Then the block set WDB to be deleted is sampled, as shown in Figure 9.

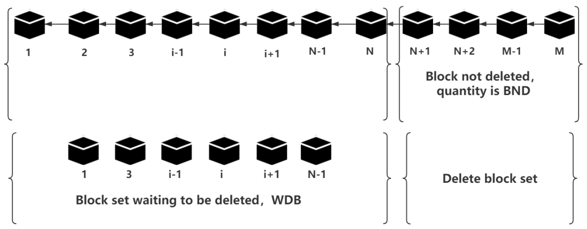

1
2
3
i-1
i
i+1
N-1
N
N+1
N+2
M-1
M
Block not deleted, quantity is BND
Block set waiting to be deleted, WDB
Delete block set

FIGURE 9. Node status of the block to be deleted.

(3) Each block is verified starting from the block creation. After the verification, block deletion is performed according to the set WDB, and the deletion information is recorded. Finally the DBS of the deleted block is obtained, as shown in Figure 10.

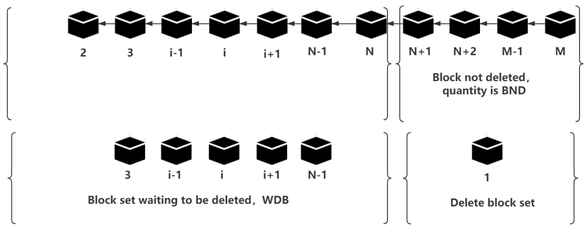

2
3
i-1
i
i+1
N-1
N
N+1
N+2
M-1
M
Block not deleted, quantity is BND
Block set waiting to be deleted, WDB
1
Delete block set

FIGURE 10. Deleting block node status.

(4) After the node completes the block deletion, the semi-full node publishes its own deleted block set (DBS) to the blockchain network, as shown in Figure 11.

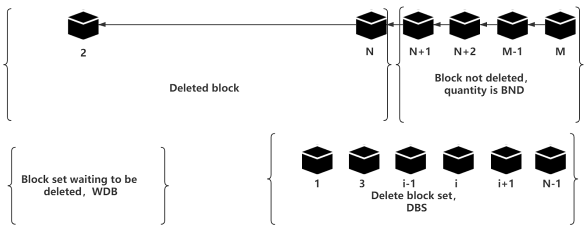

2
N
N+1
N+2
M-1
M
Deleted block
Block not deleted, quantity is BND
Block set waiting to be deleted, WDB
1
3
i-1
i
i+1
N-1
Delete block set, DBS

FIGURE 11. The status of the completed node has been deleted.

b: LIGHT NODE CONVERT INTO SEMI-FULL NODE

It is relatively simple to convert from light node to semi-full node.

Firstly, visit other nodes to confirm the status of the blockchain network. Return the block height value M, the maximum allowable block deletion probability value q, and the non-blocking block value BND.

Secondly, the MCMC algorithm with probability of q is used to obtain the random synchronous block set $RSBS$ and the continuous synchronous block set $CSBS$, as shown in Figure 12.

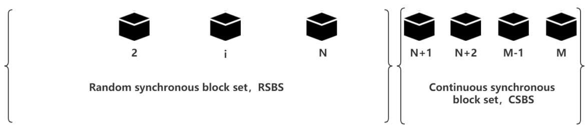

2
i
N
N+1
N+2
M-1
M
Random synchronous block set, RSBS
Continuous synchronous block set, CSBS

FIGURE 12. The state of the node to be synchronized when the light node type is converted into a semi-full node.

Again, the light nodes synchronize blocks according to the order from the creation block.as shown in Figure 13.

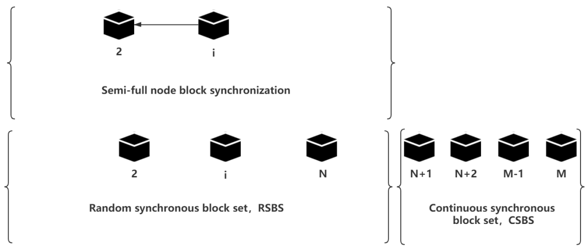

2
i
Semi-full node block synchronization
2
i
N
N+1
N+2
M-1
M
Random synchronous block set, RSBS
Continuous synchronous block set, CSBS

FIGURE 13. Node status of syncing block.

Finally, after the synchronization is completed the set RSBS is sent to the blockchain network, as shown in Figure 14. In order to let other nodes know the deleted block of the current node, it also send the elliptic curve signature and verification public key in the corresponding file.

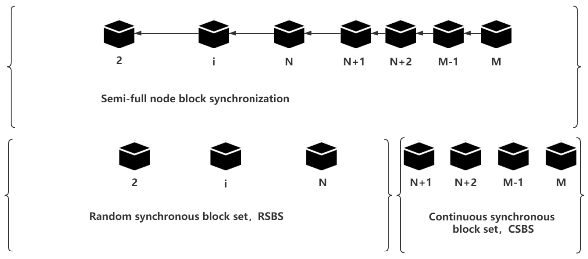

2
i
N
N+1
N+2
M-1
M
Semi-full node block synchronization
2
i
N
N+1
N+2
M-1
M
Random synchronous block set, RSBS
Continuous synchronous block set, CSBS

FIGURE 14. Node status after synchronization.

c: APPLY DIRECTLY FOR SEMI-FULL NODE

The process for node to apply for semi-full node and the process of light node type conversion to semi-full node are similar.

(I) If the node is in the consortium blockchain network, it needs to pass the review before it can synchronize the blockchain. If it is in a public blockchain network, it can directly initiate a request to become a semi-full node without review.

(II) The applicant node initiates a block header synchronization request to other nodes.

(III) After synchronizing the block header in semi-full section (in fact, it has become a light node), confirm the network status with the block network.

(IV) The remaining steps are the same as the conversion of light node type to semi-full node in 2.

C. REASONS FOR RETAINING SEMI-FULL NODES

In Bitcoin blockchain, transactions are stored in blocks in the form of ledgers [2], so each transaction's transfer will have a integrated record for input and output. After each transaction passes the verification, it will be stored in the latest block. When the transaction is used (or Bitcoin is spent), it will be recorded in the next block after passing the verification, so a long transaction chain will be formed in Bitcoin application. When the node verifies the correctness of this transaction, it will query the block of recording this transaction. After check, then it will pack the transaction into the current highest block, and finally perform ''mining game''. In fact, there are a large number of Bitcoin that are not used for the second time in Bitcoin network (Bitcoin is not used when it is written into the block). In this case, the closer the block to the current height, the more probability it will be referred and verified during transactions. Therefore, Bitcoin with frequent circulation will often appear in the recent transaction. In order to verify the reliability of this transaction, we only need to verify the latest block of this transaction, while the less commonly used transactions are often left behind over time. Based on this, we assume a block value BND that is not deleted. These blocks can verify most of the transactions that may occur next time. Of course, the value of BND will change dynamically with the network situation, rather than a fixed value. When using random algorithm to delete blocks, the threshold of ''mining game'' is reduced, and the decentralization of blockchain network is improved.

In most public blockchain networks, the proportion of a single full node in the whole network computing capability increases with the passage of time, and the greater the computing capability, the more the reward of ''mining game''. Therefore, these nodes with large computing capability will gradually control the whole blockchain network, making the network more and more centralized. This would violate the original intention of blockchain decentralization. So we define a kind of semi-full node between the whole node and the light node to reduce the proportion of computing capability of the whole node for the whole network, which make the network tend to be decentralized again.

D. SEMI-FULL NODE OPERATION LOGIC

As a semi-full node, it can not only delete the blocks at the initial stage of blockchain creation, but also can delete the recently produced blocks when the blockchain network is running.

As shown in Figure 15, when the node is running, the blocks stored on the node can be divided into three parts: the processed random block part (*BRA*), the waiting block part (*BWRA*), and the unprocessed reserved block part (*RB*).

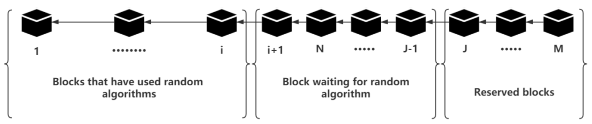

1
·······
i
Blocks that have used random algorithms
i+1
N
·····
J-1
Block waiting for random algorithm
J
·····
M
Reserved blocks

FIGURE 15. Block status in runtime node.

Block deletion conditions:

NBT refers to the block volume of the block of *RB* part for the transaction of *RB* part.

qBNDT is the total block quantity of *RB* part

g is the reference proportion of transaction reference block volume (80% used in this paper)

$$\frac{NBT}{qBNDT} \geq g \tag{6}$$

When the running semi-full node meets the block deletion condition (as shown in Figure 16), the block waiting for processing will be expanded (the block in *RB* will be moved to *BWRA*). When the waiting blocks reach 1000 blocks, random deletion processing will be performed according to the probability. If the conditions are not satisfy, each time a node synchronizes a block or generates a block by itself, it will add the block to the *RB* part (as shown in Figure 17) and calculate the conditions. The system will loop this condition all the time, so that the generated blocks are continuously deleted.

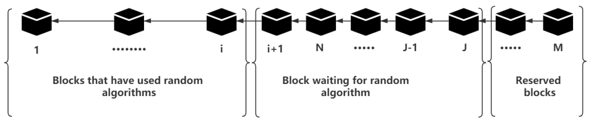

1
·······
i
Blocks that have used random algorithms
i+1
N
·····
J-1
J
Block waiting for random algorithm
·····
M
Reserved blocks

FIGURE 16. Node block state when block deletion condition is satisfy.

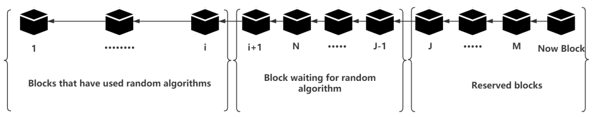

1
·······
i
Blocks that have used random algorithms
i+1
N
·····
J-1
Block waiting for random algorithm
J
·····
M
Now Block
Reserved blocks

FIGURE 17. Node block state when block deletion condition is not satisfy.

Firstly, the block part waiting to be processed is determined, and the deletion set *WDB* is obtained by random filtering. As shown in Figure 15, the blocks to be processed have been filtered in the *WDB* set.

Secondly, according to figure 19, nodes start to delete blocks in *BWRA* part according to *WDB* set, and then record the deleted blocks in *DBS*.

Finally, as shown in Figure 20, the node completes the deletion of *BWRA*, publishes the set *DBS* to the network, and also sends its own elliptic curve signature and the node's own public key.

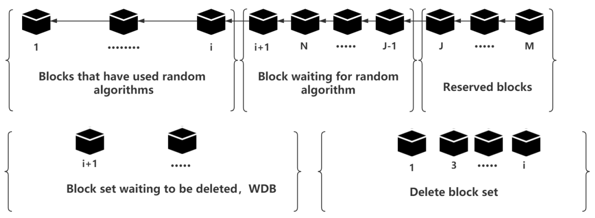

1
·······
i
Blocks that have used random algorithms
i+1
N
·····
J-1
Block waiting for random algorithm
J
·····
M
Reserved blocks
i+1
·····
Block set waiting to be deleted，WDB
1
3
·····
i
Delete block set

FIGURE 18. Initial state of node pruning block under running conditions.

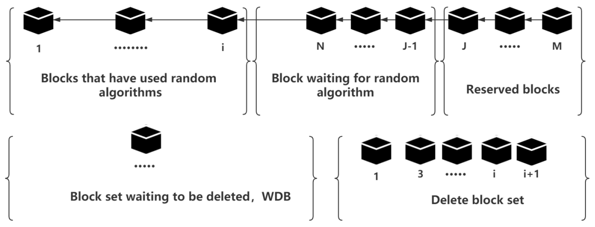

1
·······
i
Blocks that have used random algorithms
N
·····
J-1
Block waiting for random algorithm
J
·····
M
Reserved blocks
·····
Block set waiting to be deleted，WDB
1
3
·····
i
i+1
Delete block set

FIGURE 19. Block deletion by nodes under operation conditions.

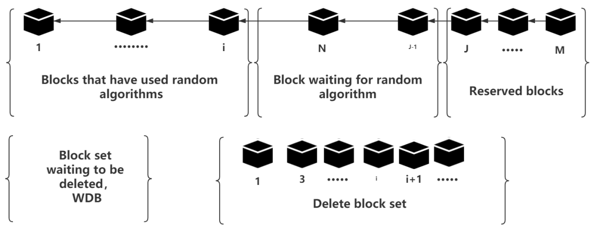

1
·······
i
Blocks that have used random algorithms
N
J-1
Block waiting for random algorithm
J
·····
M
Reserved blocks
Block set waiting to be deleted, WDB
1
3
·····
i
i+1
·····
Delete block set

FIGURE 20. Nodes complete block deletion under operation conditions.

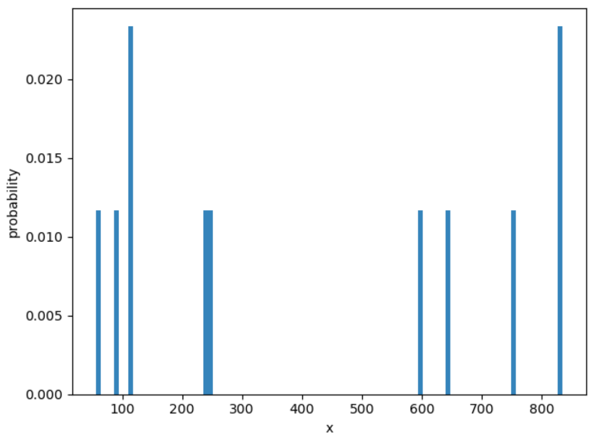


FIGURE 21. The probability of 10 nodes deleting corresponding number blocks.

V. EXPERIMENT OF RANDOM PRUNING ALGORITHM

In order to verify the reliability and stability of the random algorithm, we test and analyze the algorithm to ensure that the deleted block will not have a greater impact on the nodes.

A. INTRODUCTION TO THE EXPERIMENTAL ENVIRONMENT

The test environment of the model is as follow: the programming experiment of the win10 system in Python and go language environment.

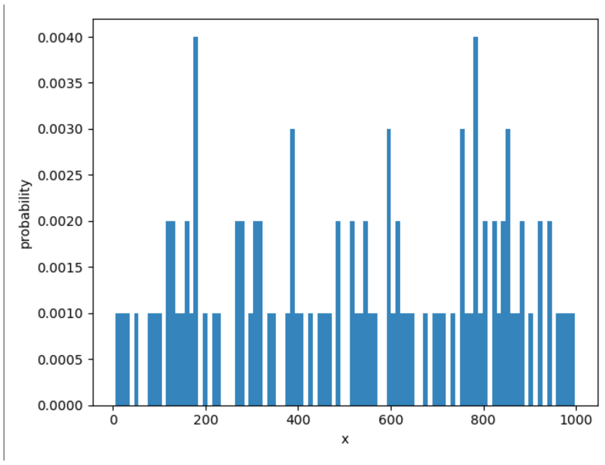


FIGURE 22. The probability of 100 nodes deleting corresponding number blocks.

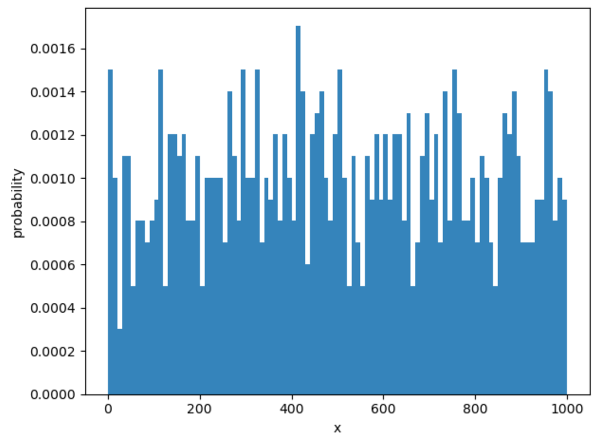


FIGURE 23. The probability of 1000 nodes deleting corresponding number blocks.

Operating System: Windows 10 64 Bit (DirectX 12)
CPU: Intel Core i7-8750H @ 2.60GHz
Memory: 16 GB (Samsung DDR4 2133MHz/Hynix DDR4 2133MHz)
Video Card: Nvidia GeForce rtx2060 (6 GB)
Language tool: pycharm and goland

B. RANDOM DELETION EXPERIMENT

Under the condition of randomly deleting 80% blocks, we do 1000 block deletion experiments on 10, 100, 1000 and 10000 nodes respectively, check the stability of deletion, and get figure 21, Figure 22, Figure 23 and Figure 24 (in the four pictures, X-axis represents the number of blocks, and Y-axis represents the probability of block selection).

In Figure 24, it can be observed that with the increasing number of nodes, the deleted blocks tend to be stable, and it can be concluded that when there are enough nodes, the probability of deleted blocks will become equal gradually. In this way, we can avoid the hidden danger of system security caused by too many or too few blocks in a certain part and ensure the integrity of blocks in the whole blockchain system.

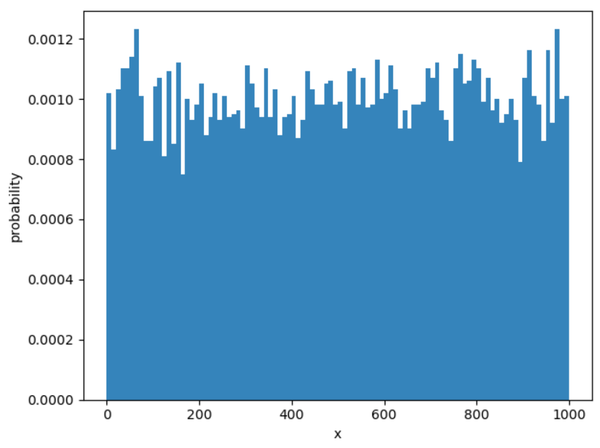


FIGURE 24. The probability of 10000 nodes deleting corresponding number blocks.

TABLE 2. Comparison of three block storage reduction schemes.

	Degree of compression	Scalable	Transaction traceability performance	Block verification capability
Semi-full node	High(70%-90%)	High	High	All
Compression Algorithm for Blank nodes	Lower(10%-15%)	Medium	High	All
Mini-blockchain scheme	Medium(20%-40%)	Lower	Lower	part

Finally, we compares Compression Algorithm for Blank nodes [28] and Mini-blockchain scheme [29] with our scheme in Table 2.

VI. SUMMARY AND OUTLOOK

This paper presents a method to reduce the storage requirement of the block chain system without affecting the security and integrity of the block chain. The data analysis proves that the segmented blockchain reduces the data requirement greatly compared with the central blockchain. Therefore, using the distributed storage strategy of blockchain to provide impetus for the development of blockchain network has more advantages.

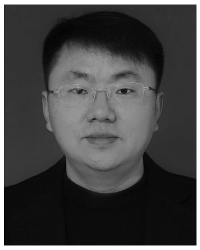
In this paper, the semi-full node is introduced and designed, comprehensive experiments are carried out to test the stability of the deleted block when 80% block information is deleted. However, due to the limitation of equipment conditions, the block probability of deletion is the estimated value, and only a simple estimation operation is done. In the future work, we will further calculate the pruning rate accurately and measure the best pruning block rate.

REFERENCES

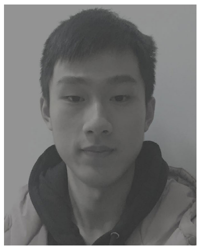
[1] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[2] *Bitcoin Cash Charts*. Accessed: Jun. 10, 2020. [Online]. Available: https://www.Bitcoin.com
[3] (2010). *Ethereum White Paper*. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[4] *Ethereum Cash Charts*. Accessed: Jun. 10, 2020. [Online]. Available: https://www.ethereum.org
[5] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable BlockChain for IoT security and privacy," 2017, *arXiv:1712.02969*. [Online]. Available: http://arxiv.org/abs/1712.02969
[6] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, 2015, pp. 919–927.
[7] A. Fitwi, Y. Chen, and S. Zhu, "A lightweight blockchain-based privacy protection for smart surveillance at the edge," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 1–8.
[8] J. Eberhardt and S. Tai, "On or off the blockchain: Insights on offchaining computation and data," in *Proc. Eur. Conf. Service-Oriented Cloud Comput.* Cham, Switzerland: Springer, 2017, pp. 3–15.
[9] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598.
[10] Y. Xu and Y. Huang, "Segment blockchain: A size reduced storage mechanism for blockchain," *IEEE Access*, vol. 8, pp. 17434–17441, 2020.
[11] S. Kim and J. Kim. *POSTER: Mining with Proof-of-Probability in Blockchain*. [Online]. Available: https://www.researchgate.net/publication/325480435_POSTER_Mining_with_Proof-of-Probability_in_Blockchain
[12] K. Li, H. Li, H. Hou, K. Li, and Y. Chen, "Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain," in *Proc. IEEE 19th Int. Conf. High Perform. Comput. Commun.*, Dec. 2017, pp. 466–473.
[13] H. Ritzdorf, C. Soriente, G. O. Karame, S. Marinovic, D. Gruber, and S. Capkun, "Toward shared ownership in the cloud," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 12, pp. 3019–3034, Dec. 2018.
[14] B. D. Guide. (2016). *Simplified Payment Verification (SPV)*. [Online]. Available: https://bitcoin.org/en/developer-guide
[15] D. Larimer, "Delegated proof-of-stake (DPoS)," Bitshare, White Paper, 2014. [Online]. Available: https://github.com/bitshares-foundation/bitshares.foundation/blob/master/download/articles/BitShares Blockchain.pdf
[16] K. D. Gupta, A. Rahman, S. Poudyal, M. N. Huda, and M. A. P. Mahmud, "A hybrid POW-POS implementation against 51 percent attack in cryptocurrency system," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2019, pp. 1–8.
[17] Y. Xu and Y. Huang, "An n/2 Byzantine node tolerated blockchain sharding approach," in *Proc. 35th ACM/SIGAPP Symp. Appl. Comput.*, 2020, pp. 349–352. [Online]. Available: http://gofun.online/document/blockchain_sharding.Pdf
[18] R. Guerra, "Likelihood, Bayesian and MCMC methods in quantitative genetics," *J. Amer. Stat. Assoc.*, 2008.
[19] G. O. Roberts and J. S. Rosenthal, "Examples of adaptive MCMC," *J. Comput. Graph. Statist.*, vol. 18, no. 2, pp. 349–367, Jan. 2009.
[20] *Bootstrap and Monte Carlo Methods in Biology*, Manly B F J. Randomization, Boca Raton, FL, USA, 1997.
[21] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *J. Comput. Graph. Statist.*, vol. 9, no. 2, pp. 249–265, Jun. 2000.
[22] Y. Zhai and M. Yeary, "Implementing particle fifilters with Metropolis-Hastings algorithms," in *Proc. Region 5 Conf., Annu. Tech. Leadership Workshop*, 2004.
[23] C. K. Carter and R. Kohn, "On gibbs sampling for state space models," *Biometrika*, vol. 81, no. 3, pp. 541–553, 1994.
[24] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic digitalsignature algorithm(ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, 2010.
[25] M. A. Andreas, *Mastering bitcoin*. Newton, MA, USA: O'Reilly Media, 2014.
[26] R. C. Merkle, "Protocols for public key cryptosystems," in *Proc. IEEE Symp. Secur. Privacy*, Apr. 1980, pp. 122–134.
[27] M. Szydlo, "Merkle tree traversal in log space and time," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Interlaken, Switzerland, May 2004, pp. 541–554.
[28] X. Chen, S. Lin, and N. Yu, "Bitcoin blockchain compression algorithm for blank node synchronization," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Xi'an, China, 2019, pp. 1–6, doi: 10.1109/WCSP.2019.8928104.
[29] J. Bruce. (2014). *Purely P2P Crypto-Currency With Finite Mini-Blockchain*, [Online]. Available: https://www.bitfreak.info/files/pp2p-ccmbc-revl.pdf

PENG ZHAO was born in 1994. He received the B.E. degree from the Zhejiang University of Technology, Hangzhou, China, in 2020. His research interest includes blockchain, networks security, and cloud computing.



HONGBING CHENG was born in 1979. He received the Ph.D. degree from the Nanjing University of Posts and Telecommunications. He is currently a Professor. His research interests include blockchain, cryptography and information security, computer communications and networks, and cloud computing.



YICHENG FANG was born in 1997. He is currently pursuing the master's degree with the Zhejiang University of Technology, Hangzhou, China. His research interest includes blockchain, networks security, and cloud computing.

XIAOQING WANG was born in 1995. She is currently pursuing the master's degree with the Zhejiang University of Technology, Hangzhou, China. Her research interest includes blockchain, networks security, and cloud computing.