2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications

MWPoW: Multi-Winner Proof of Work consensus protocol

An immediate block-confirm solution and an incentive for common devices to join blockchain

Yibin Xu
School of Computer Science and Informatics
Cardiff University
Cardiff, UK
work@xuyibin.top

Yangyu Huang
School of Electronic Engineering and Automation
Gulin University of Electronic technology
Gulin, PRC
i@hyy0591.me

Abstract--**In this article, we present designs for a Multi Winner Proof of Work (MWPoW) consensus primitive. MWPoW is a consensus protocol that places the group mining idea directly into a protocol as to prevent power centralization, higher general reward expectation among participants and shorten the interval time of block generation. It makes every miner a component of one of three groups assigned by the network, not as an individual competitor; all users in the winner group receive compensation. The hash difficulty is higher than ordinary blockchain because there are only three parties in the system. MWPoW powered blockchain is more secure from tampering. The blocks can be quickly final accepted without later block confirms, which makes it possible for block interval time to be minuteness. The experiment suggests the hash rate of MWPoW is around 1500 times larger than Nakamoto blockchain while it only requires about 1 second to finally accept a block in the lab environment (with 3-second block interval). MWPoW is universal profitable (the mining remuneration is much more distributed than Nakamoto blockchain).**

1. Introduction

The primary concern regarding the distributed consensus of blockchain is the trend of calculation power centralization caused by the high threshold for individuals to profit from the game of proof-of-work (PoW). The fierce competition has lowered the interest of vast users to join in mining directly, stealthily blocked most individuals from the job of tamper-detect and voting but force them to hand over their personal interests to the protection of people with considerable calculation ability. To fulfill the initial idea of blockchain as decentralized and equal, the consensus-reaching approach needs to be improved. Although many kinds of research toward PoW 's alternative approach have been proposed since Blockchain technology came into public attention, researchers seldom discuss the prevention of power centralization. And the design that there is only one 'winner' elected every time is not changed, the winning possibility for an individual participant is decreasing with the grows of participants. To profit from the game, join a mining pool seems to be the solution for common users, in which, users contribute their calculation power to a mining pool, the pool uses the joined power to mine and returns compensation to participants based on their contributions. However, the security and equality can be harmed as the mining pool is a centralized model. If the owner of the mining pool is a Byzantine, it might conduct selfish mining [6] which harm the equality of blockchain by depriving others' winning opportunity. The owner of the mining pool might not return the amount of compensation that is matched to the users' contribution; The pool might be used to conduct merge-mining [7] without users' knowledge or even be used to attempt tampering. Though there is a decentralized Mining Pool [8] proposed, the funding generated needs to be frozen for a very long time in these pools, and the users in the pool still cannot make a judgment to blocks.

Another concern toward consensus protocols is the slowness of global consensus reaching. It requires several later block-confirms to determine if a block is final accepted, which is time-consuming. The time is also consumed through data propagation, which is increased with the growth of transactions and is eating into the travelling through unregular network structure. In many blockchains, the block size is set to be a rather small number to make the time of data sync acceptable. As a side-effect of that, it deteriorates the consensus scalability. Bitcoin suffers a consensus latency of about an hour (for the recommended 6-block transaction confirmation), and with up to 7 transactions per second peak throughput, only a fraction of centralized finance systems like Alipay [10] and VISA at 256,000 transactions per second and 50,000 transactions per second respectively [10]. Many Off-chain approaches are proposed [4] to respond to these shortcomings of the blockchain, but users either must continuously monitor the network to make sure that other sides are not compromising them in the payment channel or they must surrender their private key to the custody of third parties. Another kind of approach to scaling the blockchain is to simplify the transactions a block contained. It is of the high possibility that most nodes have already received the majority transactions included in a block before receiving this block, and they only need to determine if the transactions in the block have been in their MemPool. Thus, necessarily, a block is only required to include the IDs of transactions. There are blockchains which use this approach, e.g. Corallo's Compact block [3] and Xtreme Thinblocks [9]. However, as the block simplify approaches are still requiring a long pending time; there are still rooms for improvement.

In this article, we propose Multi-Winner Proof of Work (MWPoW), which protocol intends to solve blockchain's shortcomings concerning power centralization and the inefficiency of transaction confirm while enable nodes with disadvantage in calculation ability to easily profit from the mining game . Before joining the mining game, nodes in MWPoW need to claim the amount of calculation power it intended to put into every round of the mining game and showing evidence as the prove. When the evidence is embedded into a block, a try range will be assigned to the node. The node can then try to create a block and find a Nonce in this try range. There are two difficulties of a block, entrance difficulty and accept difficulty. When a node finds a Nonce that fulfills the entrance difficulty it will broadcast the block with the Nonce to the network, miners of the same group with this node will try to find Nonce of this block in their try ranges which fulfill the accept difficulty. When which is located and broadcasted to the network, this block is announced. During the announcing, a miner will broadcast Nonces which do not fulfill the accept difficulty but fulfill at least 25% of the power it claimed before. The Nonces sent will be embedded into the next block of this announced block, and the remuneration will be given based on the Nonce miners sent during the announcing at the next block of this announced block. Nonces cannot be stolen because nodes are overseeing different try range. Some procedure will be conducted to expel unquantified miners and to reassign try ranges for valid miners after every round of the game. These procedures, as well as the procedure for forming groups, will be discussed in section 3.

We take the idea of simplifying blocks into the design MWPoW; MWPoW is built up with IBLT (Invertible bloom look-up table) and Bloom Filter. The additional information like Nonces and the evidence of power claims which are required to be embedded into blocks will not largely affect the scalability of blockchain after

978-1-7281-1141-4/18/$31.00 ©2018 IEEE
DOI 10.1109/BDCloud.2018.00141

simplifying block. The reduce and the calculation of block size will be given at section 3.2.6. In section 4, we prove that compared to Nakamoto Blockchain (Bitcoin) of around 8000 nodes worldwide, miners in MWPoW only need an additional download bandwidth of 2 Kbytes/s to function the MWPoW protocol, which is almost negligible in nowadays networks.

In section 4.2 and section 4.3, we will prove that in a completely decentralized environment, MWPoW reduces the long pending time for final confirm a transaction and significantly higher the reward expectation for individual nodes who join in the mining game directly. In particular, we enable nodes to quickly determine if the majority has accepted a block without later block confirms; we design that there will be one-third of the valid miners to receive compensation base on their contribution in every round of the game, we enable devices with a low calculation ability to judge blocks and to mine as groups. Which design does not require an extended pending period like P2P mining pool [8].

In summary, we alleviated the power centralization by eliminating the gap of profiting between outsourcing calculation power to mining pool and joining the mining game directly. The design of using simplified block enlargers the consensus scalability while the design of embedding Nonces of the previous block into every block helped to determine the acceptance rate of a block quickly. These fulfilled the two requirements of immediate confirm --- transactions are quickly embedded into a block, and the block is soon being finally accepted. Besides, the security of blockchain is reinforced, the experiment in section 4.3 suggests that the hash rate of MWPoW is around 1500 times larger than Nakamoto blockchain while it only requires about 1 second to finally accept a block in the lab environment (with 3-second block interval). Though the hash rate is largely increased, MWPoW compared to other blockchain is, in fact, energy friendly, and the block interval time can safely be reduced significantly in MWPoW, the discussion about these will be given in section 5.

We organize the paper as follows:

(1) Some basic concepts of mining, as well as Bloom filter and IBLT, is given in section 2.

(2) Section 3 presents a description of MWPoW.

(3) Section 4 shows an experiment of MWPoW regarding the consensus latency, scalability as well as the hash difficulty.

(4) Section 5 shows some common concerns and answer about MWPoW.

(5) Related works are displayed in section 6.

(6) The paper is concluded in section 7.

2. Preliminaries

2.1 Difficulty, Mining Pool and Pay Per Share

Blockchain systems usually have a global target for generating a valid block. The difficulty is a measure of how difficult it is to find a hash below the given global target current_target. [11] Difficulty is defined as difficulty $= \frac{\text{difficulty\_1\_target}}{\text{current\_target}}$, where difficulty_1_target is 0x1dffff ($0x00ffff * 2^{8*(0x1d-3)}$ in hex). A mining pool is a blockchain node that instead of finding the hash itself, it asks its power-sharing partners to find the hash in different try range concurrently. When the solution is found, the compensation will be given to the participants by mining pool through transfer base on the work they contributed. To measure the work participants done, mining pool sets sub-targets of the global target which is much easier to fulfill when attempting to find the one that fulfills global target. Hashes which fulfilled the sub-targets are called shares. Pay per share is a mining pool clearing scheme, the amount of compensation one will have is set base on the number of shares it submits.

2.2 Bloom filter and Invertible bloom look up table

Bloom filter (BF) is an array which can represent n items through m bits of data. All m bits of data are set to be negative at first. When an item is inserted into the filter, k bits in the array, which is selected using k hash functions will be set to be positive. $\mathrm{m} = -\mathrm{n}log_2^{(f)}/ln^2$, f is the intended false positive rate (FPR). An item is inside the bloom filter when all k bits of data is positive. Invertible bloom Look Up Table [5] is an extension of BF that store key-value pairs and allow the recovery of the original data set. An IBLT stores a set of key-value pairs (x,y) in an m sells table (If the keys or values are not number and can be represented by fixed-length Bit-strings, they can be transferred into one by taken to XOR). There are three fields in each cell, which are defined as follows:

Table.1. Fields in a sell of IBLT

Field name	Description
KeySum	the sum of the keys x that have been added to the cell
ValueSum	the sum of the values y that have been added to the cell
Count	the number of pairs that have been added to the cell.

The following operations are defined over an IBLT:

Table.2. Operations in IBLT

Operation	Description
Insert (x, y)	add a key-value pair (x, y)
Delete(x, y)	remove a key-value pair (x, y)
ListEntries	retrieve and list all the key-value pairs stored in the IBLT.
Get (x)	return the value y paired to x. Return null if there is no value paired to the key x. returning "not found", in which case there may or may not be a value associated with the key x.

3. MWPOW: Multi-Winner Proof of Work Consensus Protocol

The procedure of MWPoW are as following, the throughout discussion are in sub-sections of this section.

Prepare stage

Prior to mining, a new participant is required to claim the difficulty of crypto-puzzle it is intended to solve per round of the game and showing evidence as to prove its ability. This evidence includes the difficulty of crypto-puzzle it intended to solve per round of the game, the wallet address for receiving compensation and a Nonce that can make the Hash of this evidence fulfil the difficulty it intended to solve. After a block embedded this information, this participant is assigned into one of three different groups and is given a try range which matched to the difficulty it claimed (will be discussed in following section 3.2), it can then join in the mining game. Figure.1 shows an example of the prepare stage. When Block X-3 is the latest block in the mainchain, a new participant creates an evidence base on Block X-3. After building the evidence, the participant then sends the evidence to the network when Block X-2 comes out (when the participant finished building the evidence, it is with a significant possibility that Block X-2 has just become the latest block because the time needed to create an evidence should be similar to the block interval). Block X will embed the evidence, and the participant will be given a try range afterward. (The reason the evidence is written in block X not block X-1 is that, the succeeding round of mining game starts immediately after receiving an announced block, Block X usually records information submitted between the time interval of Block X-2 to Block X-1 while Block X-1 records information between Block X-3 to Block X-2).

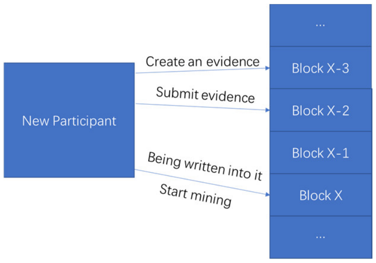


Fig 1. Prepare stage

Mining stage

There are two hash difficulties in MWPoW, entrance difficulty and accept difficulty. When a participant created a block and found a Share of this block in its try range which fulfilled the entrance difficulty, it will broadcast the block into the network. Nodes which are in the same group with this node will then try to find a Share of this block that fulfilled the accept difficulty in their try range. When a Share of accept difficulty is found and broadcasted to the network, the block is successfully announced. During the mining process, every node will broadcast the Shares it found that are below the accept difficulty but higher than 25% of the difficulty it claimed as to proof the amount of its work. The Shares cannot be stolen because every node has different try range. Figure 2. shows an example of the mining stage.

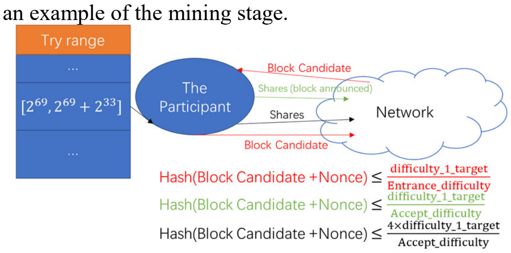

Try range
...
$[2^{69}, 2^{69} + 2^{33}]$
...
The Participant
Block Candidate
Shares (block announced)
Shares
Block Candidate
Network
Hash(Block Candidate +Nonce) ≤ $\frac{\text{difficulty\_1\_target}}{\text{Entrance\_difficulty}}$
Hash(Block Candidate +Nonce) ≤ $\frac{\text{difficulty\_1\_target}}{\text{Accept\_difficulty}}$
Hash(Block Candidate +Nonce) ≤ $\frac{4\times\text{difficulty\_1\_target}}{\text{Accept\_difficulty}}$

Fig 2. Mining stage

After announced

The group which first succeed in announcing a block will gain compensation, the compensation is given to the miners of that group directly in the Coinbase of the next block which mined on top of this block. A block will embed the Shares of three blocks of its preceding block height. The three blocks consist of the winning block (the group of miners of which block will gain the compensation), and the two blocks which have the highest hash difficulties within their groups. For the wining block, the amount of compensation of different participants is varied by the Shares they sent during the block announcing. After a block is announced, the group-assign for miners which min on top of this block will be adjusted through adding newly claimed nodes and expelling unqualify nodes, which brings a shift of assigned try range for most nodes. Thus, every block comes with a new power grouping scenario for the system. Figure 3. Shows an example of the procedures after a block is announced.

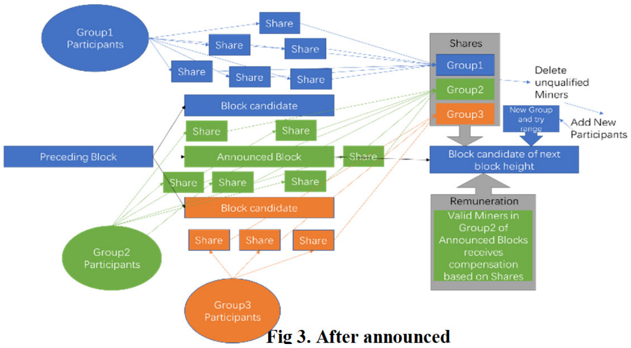


Fig 3. After announced

Branch choosing

Same as Bitcoin, miners will choose to min on the valid block which they believe to be the first block reached an accept difficulty. However, miners will change branch when there is a block which is in the highest branch and the shares in the branches stem from it altogether weight more than 50% of all the shares created since its height. This waived the necessity of pending blocks. Figure.4. shows an excessive example of the branch choosing (The "winner" can usually be determined before the next block comes out in reality). The "winner" becomes a final accepted block because the branch steams from it (blocks in purple dashes) weight more than the combination of all the shares in orange dashes and green dashes, and it is in the highest branch.

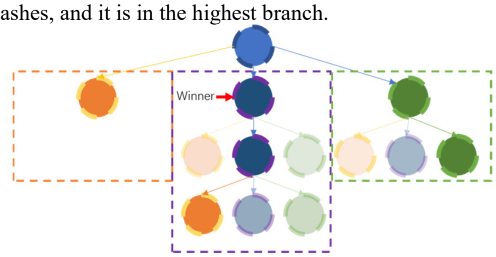


Fig.4. Branch choice

3.1 Power announces

Each node will announce the hash difficulty they are planning to place into the game and sending the evidence to the network to make participants aware of the ability of each other as to its capacity. The evidence sent is called NJ (New Join), the structure of New Join is indicated in Table.3. By sending New Join, the new miner has claimed the breach of the blockchain it chose, and the New Join is only valid until the next block comes out.

Table.3. Structure of New Join

Filed	Purpose	Size bytes
HashPrevBlock	256-bit hash of the preceding block	32
Intended_difficulty	Intended hash difficulty	4
Wallet address	Use for receiving compensation	34
Nonce	Hash tried (256-bit number, starts from 0)	32

For a New Join NJ to be valid, it should fulfill the condition that $\text{Hash(NJ)} \leq \frac{\text{difficulty\_1\_target}}{\text{Intended\_difficulty}}$, Intended_difficulty is the one indicated in New Join.

3.2 Block, group and try range assign

A block in MWPoW has four sessions. Besides the Block Header and Transactions which inherit from original blockchain structure, it has an assembly of Proof of Contribution toward Forming the Preceding Block (PC); and a session of sets of New Join (NJs).

The block header in MWPoW consists of the hash of the previous block, three MerkleRoots, entrance difficulty, accept difficulty, a bit flag and a timestamp. Table 4 shows the purposes of these fields. We do not write the Nonces of a block A directly in itself. Instead, the block of the next block high will carry these Nonces provided A has the most shares in the group which purposed it, or A is the winner block.

Table.4. Block header

Filed	Purpose	Size bytes
HashPrevBlock	The hash of the preceding block	32
TxnMerkleRoot	Merkle Root of Transactions	32
NJMerkleRoot	Merkle Root of New joins	32
PCMerkleRoot	Merkle Root of Proof of Contribution Toward Forming the Preceding Block	32
EDifficulty	Entrance difficulty	4

ADifficulty	Accept difficulty	4
Group	Indicate which group this block belongs to	1/4
Timestamp	A timestamp recording when this block was created in MS since 1970-01-01 T 00:00:000 UTC	8

PC contains three Sub-sections; each sub-section stands for a block candidate of the last block which proposed by different groups. There are three fields in a Sub-section, they are shown in Table.5.

Table.5. Structure of Sub-section of Proof of Contribution

Filed	Purpose	Size bytes
InitialHash	The block header hash of a block candidate (Without HashPrevBlock field)	32
Winner	A bit stands for if the block candidate in this sub-section wined the mining round mining game.	1/8
Shares	The set of Nonce that are submit to the network to try to make this block candidate accepted	Vary

In order to make the Sub-section valid, every Nonce in Shares field mush fulfil the condition that $\mathrm{Hash(InitialHash + Nonce)} < \frac{4\times\mathrm{difficulty\_1\_target}}{\mathrm{Intended\_difficulty}}$, supported Nonce is in a miner ‘s try range and the power it declined previously is $\mathrm{Intended\_difficulty}$. The entrance difficulty and accept difficulty for block x is:

$$AcceptDifficulty_x = \frac{\mathrm{T}\times AcceptDifficulty_{x-1}}{(timestamp_{x-1} - timestamp_{x-2})},$$

$EntrancetDifficulty_x = \min(\frac{N_{x-1}\times EntrancetDifficulty_{x-1}}{3} \times N_{x-1}, \frac{1}{2}AcceptDifficulty_x)$,where $AcceptDifficulty_{x-1}$ and $EntranceDifficulty_{x-1}$ is the accept difficulty and entrance difficulty of the preceding block of block x respective, T is the desired block interval time in MS; $timestampx$ is the timestamp of block x; N_{x-1} is the number of block candidates announced in the block height x-1 of the same block branch with block x. We set the desired number of block candidates proposed by every group to be one, thus, three candidates in total.

3.2.1 *Forming PC*

When a miner M is drafting a block, it will include three block candidates into PC which block candidates were purposed by three groups and were competed to become the preceding block of this new drafting block. Except for the winner, other candidates should be the ones which have the most Shares within their groups at the time when M is drafting this new block. A miner can only send up to 4 Shares to the network for a block candidate, if it sends more, M will randomly pick four from the shares it sent and write the Nonce in these shares into PC. A share is sized 36 bytes and has two fields in it.

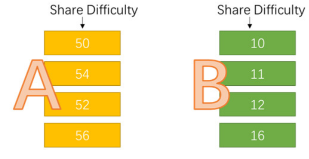
Table.6. Structure of Share

Filed	Purpose	Size bytes
L_4_D_IH	Last 4 bytes of InitialHash	4
Nonce	Hash tried (256-bit number)	32

M should mark the block candidate which, to M ’s knowledge, first have a share of accept difficulty as the winner for last mining competition period. And this group of miners will receive compensation.

3.2.2 *Mining reward*

According to the Shares in the winner Sub-section of PC, the compensation will be given to the miners in winning group directly through Coinbase base on the total reward amount of last block height multiply the sum of the miner ‘s valid share proportions to the sum of all the shares in this group. Figure.5 shows an example of this, where the sum of the difficulty of the shares sent by the Miner A and Miner B are 212 and 49 respectively, and the sum of the difficulty of all the valid shares in this group is 1000. The total reward amount of last block height is 100. Miner A and Miner B receives 21.2 coins and 4.9 coins respectively.



Transactions	
From	To
System (Coinbase)	
$100 \times \frac{212}{1000} = 21.2$	Miner A
$100 \times \frac{49}{1000} = 4.9$	Miner B
	Miner ···

Fig 5. Reward assign

3.2.3 *Participant expels and Derived Group*

If the Shares sent by a node within a round of mining game are 50% lower than the difficulty it claimed, it will be expelled from the game and it will not receive compensation even if it is in the winning group. If a node has been in the game for 500 rounds, it will also be expelled. Expelled node is returned to the prepare stage. We refer groups which have expelled unsatisfied miners as derived groups.

3.2.4 *New Joins and Try range assign*

M will include a set of valid New Joins into this new block. The valid New Joins are those which the HashPrevBlock in them is the hash of the third preceding block of which block that M is drafting. Figure.6 shows the example of this, supported M is drafting block D and the green cycles are New Joins which the HashPrevBlock in them is the hash of block A. M should include green cycles as new participants.

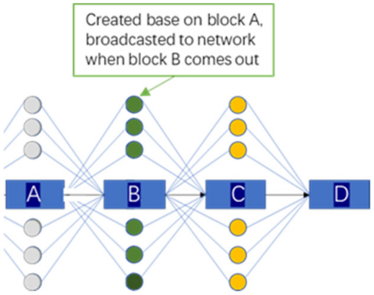

Created base on block A, broadcasted to network when block B comes out
A
B
C
D

Fig 6. Valid New Join choose

The New participants should be ranked from difficult to easy by the intended difficulty they claimed. The New Joins set will be split into three parts following the ranked sequence. The part with the biggest power will be combined with the derived group which has the smallest power remained, and the smallest part of New Joins set will pair the derived group with the biggest power, the middle one will pair the middle one. The combined groups will become the new groups in the next round of competition. Wallet accounts in the new groups will be ordered in alphabetical order and the try range will be assigned following the wallet account order. We say there are K_n miners claimed X_n amount of power in group N, $\mathrm{N} = \{1,2,3\}$, X_{n_i} is the power which miner i, $i \in K_n$ will put into the game; $A_{N_i}$is the try range that has been assigned in group N before i’s try range. The try range of i is $[A_{N_i} + 1, A_{N_i} + Try_{N_i}]$, where $A_{N_i} = A_{N_{i-1}} + Try_{N_{i-1}}, A_{N_0} = 0, Try_{N_0} = 0$; $Try_{N_i} = \frac{X_{n_i}}{X_n} \times 2^{256}$. After adding New Joins, it must guarantee that for every i, $i \in K_n$, $Try_{N_i} \geq X_{n_i} \times 2^{32}$, and the size of the encoded block is smaller than 100KBytes (the size of the block will be discussed in section 3.3). If it is not fulfilled, M should cut the number of New Joins from the tail of the ranked sequence. Figure.7. shows the structure of block and procedure of group assign.

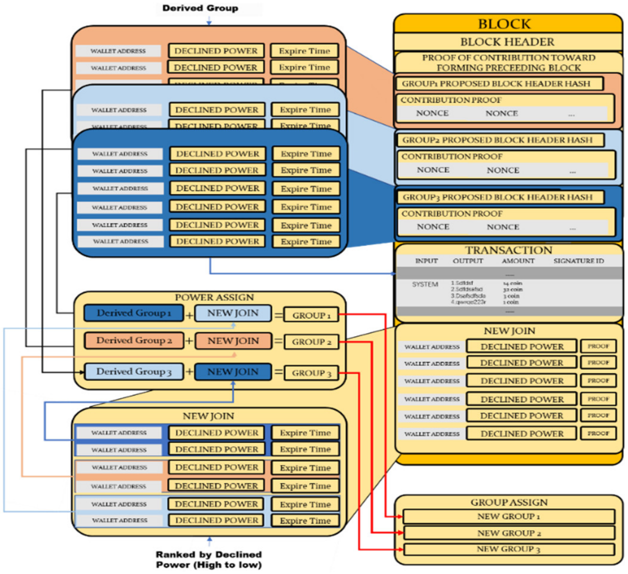


Fig 7. Structure of block and procedure of group assign

3.2.5 Block verification

After drafting the block, M will try to find a Share of this new block in M 's try range which the difficulty of the Share stands for is at least entrance difficulty. If such a share is found, M will broadcast the block and the Nonce to the network. If this block is valid, all the miners in the network will temporarily keep this block until a block in the same height becomes final accepted. If other miners in the same group with M recognized M 's block as a valid block, they would continue to try to find a share of M 'block that fulfils the accept difficulty (the group here, we referred to, is the one assigned by the preceding block of M 'block). If there are blocks of the same group which already fulfilled the entrance difficulty, the group members will still try to find shares of the block M created until there is another block in this group which has received shares from half of the group members. Group members will then drop other block candidates in this group, and to focus on finding the shares of that block. A miner is a valid block candidate for a miner Bob, when and only when:

(1) The transactions in it are correct.

(2) Items in NJ and PC are valid.

(3) Items in NJ and PC should at least 90% the same with the items Bob received in advance. For example, if Bob knows there should be K New Joins add into the new block, however, this block only contains 89% of the New Joins in those K New Joins, this block is invalid, if Bob received K shares in advance, this block only contains 89% of them, this block is invalid.

3.2.6 Encode block using BF and IBLT

To reduce the enlarged size of the block which caused by recording NJ and PC; to add more transactions into a block, all transactions, NJ, and PC in the block are encoded using an encoding method that combined BF and IBFT. An encoded block is a block with a block header and three BFs and three IBFTs which are created using the involved transactions, NJ, and PC respectively. The encoding method we used is edited from graphene [12]. When received a block, the receiver will first try to pass all the transactions, NJ and PC it had through these three BFs respectively in the block and using the succeed passed items to form IBFTs. The block is decoded correctly if the IBFT the receiver formed is identical to the ones received and the Merkle roots formed by the items decoded through the IBLTs received are identical to the Merkle Roots indicated in the block header. Unlike previous encoding methods which only use IBFT, graphene never sends an explicit list of transaction IDs; and unlike methods which only use Bloom Filter, Block Filter in graphene is much smaller, it allows FPR to be high because IBFT will find out any mistakes made. When the subtractions of two IBLTs are found, the receiver will form a BF and an IBLT use partly decoded items and send it back to the sender, then the sender will determine the subtractions and provide more information regarding the items the receiver doesn't have which resulted in the failure of decoding, the size of which information is still small presumably the receiver was actively receiving all the broadcasted information in this network. Table. 7. Shows the summary of the procedure of encoding and decoding of the block. The false-positive rate for the block filter in our method is $f = \frac{b}{r}$, where b is the expected symmetric number of differences between the IBLT in the block and the IBLT the receiver created using items passed through BF; r is the expected number of items which the receiver has but not in the block, r (r>0) is set to be m-n at the first time, where m is the number of items in respective session of Mempol and n is the number of entries in respective session of the block, every time a new block comes, after decoding, r for the next time will change accordingly, $r = average(count(I_m \cap I_n), r)$, I_m is the items in Mempol and I_n is the items in block. b is the main affecting element for the number of cells in IBLT, if the number of cells for each of these two IBLTs is d times of b, the two IBLT can be decoded by each other with a high probability [13], we set d = 1.5. In our approach, a cell in IBLT sized 10 bytes which is composed of CountSum (2 bytes), Key (4 bytes) and ValueSum (4 bytes). The ValueSum contains the last 4 bytes of the sha256d hash of the items the cell stands for. The size of the IBLT with b entries symmetric differences between the one in block and the one the sender received is 15b bytes. Thus, the space demand for this combined method in bytes is $S(b) = -\frac{\mathrm{n}log_2^{\left(\frac{b}{r}\right)}}{8ln^2} + 15b$. We re-write, as the actual implementations involves non-continuous ceiling functions:

$S(b) = \frac{1}{8}(\ln\left(\frac{r}{b}\right)[\frac{\mathrm{n}ln\left(\frac{b}{r}\right)}{[\ln\left(\frac{b}{r}\right))]ln^2(2)}) + 15b$. The lowest value of b can be found by a brute-force for-loop search at little perform cost. Thus, the total space demand for the block encoded in this method is $SIZE_{InitialHash} + S(b_{PC}) + S(b_{NJ}) + S(b_{TRANSACTIONS}) = 144 + S(b_{PC}) + S(b_{NJ}) + S(b_{TRANSACTIONS})$, we set the size of the encoded block to be 100 Kbytes maximum. The procedure of encoding and decoding are shown in table.7.

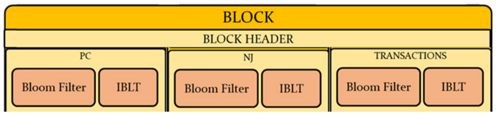


Fig.8. Encoded block

Table.7. A summary of Encoding and Decoding Block

Sender/Receiver	Purpose
Sender	Sends inv for a block and the number of entries in PC of this block and the number of entries in NJ of this block and the number of transactions in this block
Receiver	Requests unknown blocks; includes three r (the expected number of items which the receiver has but not in the block.)
Sender	Sends Bloom filter and IBLT (each created from the set of PC, NJ and transactions in the block) and header
Receiver	Pass the items it has through the bloom filters and create IBLT

4. Bandwidth Demand and Experiment

Encode the block using graphene has largely reduced the size of the block, however, to verify the block, miners still needed to hear NJs and Shares before receiving the block, and which information is eating the bandwidth. An NJ takes 102 bytes of data while a Share takes 36 bytes of data. Figure.9 shows the amount of data needed for hearing NJs and Shares with the number of valid miners in the network. Every miner sends 4 Shares to the network per round of the game, and in every round of the game we add 200 new miners into the network (The NJ of these miners will be embedded in the block of this round of mining game) until there are 10000 miners.

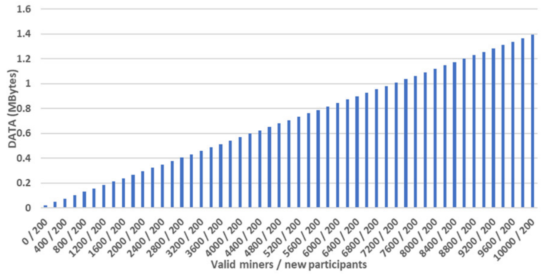


Fig.9. the amount of data for NJ and Shares VS the miner number

Bitcoin nowadays has a steady number of around 8000 miners worldwide in the network. Let's assume MWPoW also has this user scale, the minimum bandwidth for an individual node will be $\frac{1.12}{\text{Interval}}\ Mbytes$ where Interval is the predetermined block interval (in minute). The minimum bandwidth with different block interval is shown in Figure.10.

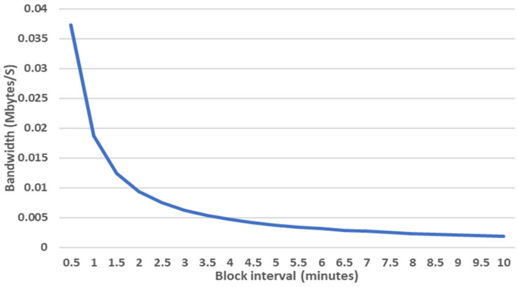


Fig.10. the minimum bandwidth for NJ and Shares VS block interval

4.1 Experiment Setup

The encoding method of MWPoW is developed from Graphene. Graphene is strictly more efficient than Compact Blocks unless the set of unconfirmed transactions held by peers is 1,287,670 times larger than the block size (e.g., over 22 billion unconfirmed transactions for the current Bitcoin block size.) [12]. Since NJ and PC which we added can be seen as additional transactions of small size, the performance of Transactions per second of MWPoW is more about Graphene than our improvements. Thus, we do not conduct the comparison of Transactions per second with other blockchains in our paper, because the comparison has been done in [12], and Graphene can process the most per second.

The purpose of our experiment is to survey the time required for nodes to finally accept a block in the different average bandwidth of the network and how easy it is for miners to receive compensation. Since other blockchain platforms are using a fixed number of later block confirms to determine if a block is final accepted, which is a steady time window, the comparison of the time for final accept a block between different block-chains is of little meaning. However, to show the security of MWPoW, we will compare the hash difficulty of accepted blocks of different blockchains with the same block interval time setup. We expect the experiment to show the result that finally accepts a block in MWPoW requires minuteness of time and does not require later block confirms while the hash difficulty is much higher than other blockchains.

We use an emulated network with 2,000 Section-Blockchain nodes. 2,000 ubuntu systems are running as VPS (Virtual Private Server) on eight HP ProLiant SL230s Gen8 server; each VPS runs a node of Section-Blockchain. Hereinafter, nodes we referred to is implemented all the elements introduced in the above section. We use a basic emulated network with 2,000 nodes, 38096 number of connections (19 connections per node in average) and 50 MS delay time per connection on average. Figure 11 and 12 shows the distribution of the number of connections and the distribution of connection delays. This basic network is the same for all following experiments conducted. In different experiments, three different full-duplex bandwidth schemes of 1 Mbytes, 5Mbytes, 10Mbytes per node in average are put into this basic network, the statistic information of three bandwidth schemes can be found in Figure 13.

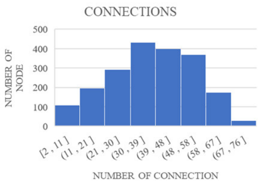


Fig.11 Connection per Node Overview

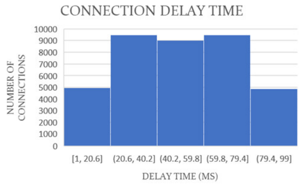


Fig.12 Delay Time per Connection Overview

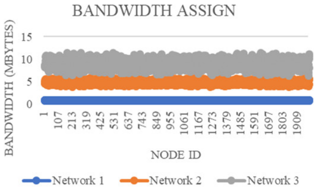


Fig.13 Bandwidth Assign

4.2 Average include rate and transaction pending time

We conducted nine experiments, the experimental setups of which are the full permutations of three bandwidth schemes introduced above with the three constant transaction traffic schemes of adding 1,000, 10,000 and 40,000 transactions per second from random nodes into the network (the transactions are sent in the complete distribution of time within every second). Three to six nodes (at least one node from each group) are selected by the algorithm to generate a block close to the end of the block interval; every experiment lasts 10 minutes. The size of transactions used in the experiments is set to be 369 bytes (the size for a transaction with an input address and an output address), and there is no limit on transactions per block. The block interval is set to be 3 seconds (Thus, about 85 Kbytes/s are used for NJs and Shares). The comparison of results among the experiments is shown in Figure 14, Figure 15 and Figure 16. Figure 14 shows the average of the percentage of transactions ever embedded into blocks of mainchain VS all the transactions ever sent to the network until receiving a new block. As can be seen from the Figure 14 and Figure 15, in average, when a block is published, more than 99.8% of the transactions ever existed in the network has been confirmed in the network situation of the bandwidth 'network 1' and a constant transaction traffic of adding 1000 transactions per second into the network. While with the bandwidth 'network 2', and 10000 transaction traffic, in average, every time a block is published, also, more than 99.8 % of the transactions ever existed in the network has been confirmed. The average bandwidth of these two networks are well covered the data capacity required per second (1000*369bytes =0.35 Mbytes, 10000*369 bytes =3.5 Mbytes), the remaining 0.2% unconfirmed transactions are the transactions created during the block broadcasting, which will be written into the next block. Thus, if the average bandwidth is well covered the data capacity required, there will be no delay to write a new transaction into the block.

However, with the bandwidth 'network 3' and 40000 transaction traffic, more than 71% of the transactions confirmed and leaving 29% of the transactions in the pending situation. It is observed in the experiment that the pending transactions would be embedded to a block after 29 blocks since it been sent to the network in average (about 1 minute). While with 40000 transactions, in network 2 and network 1, the pending transactions will be added to a block after 64 blocks and 93 blocks in average (around 3 minutes and 4.5 minutes since sent to the network). Because we add a fixed number of transactions per second to the network. when the capacity per second doesn't fulfill the data required, it is always not fulfilled during the experiment. Thus, the pending time is slowly increasing. However, in the real world, it is much more likely that there is the busy period and the spare period, the pending time will increase and decrease with the data flow as like traditional blockchains.

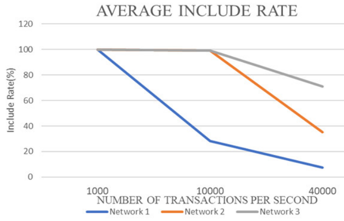


Fig.14 Average include rate in different network and number of transactions

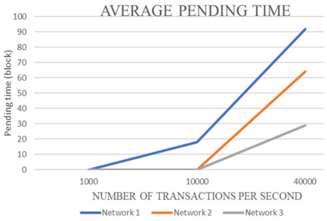


Fig.15 Average transaction pending time

$$\text{Include rate} = \frac{\textit{Number of transactions embeded in the blocks of the mainchain as of receiving}}{\textit{Number of transactions ever sent as of receiving}}$$

The transaction per second ability of MWPoW is powered by Graphene, since it has been proved in [12] that Graphene has the best transaction per second ability, other blockchains with a small number of fixed upper limit of transactions per block and does not lightweight blocks are naturally to have a much lower average include rate than us in the same experiment setup. However, original Graphene is slightly more efficient than MWPoW regarding average include rate because given the same amount of transactions, MWPoW needs to additional process NJ and PC, which taken specific bandwidth. However, to use a transaction safely, the block embedded this transaction must have been final accepted. Figure 16 shows the time for MWPoW to finally accept a block in different network bandwidth in average. As can be seen from the result, the difference of the final accept time is small. At worst, to final accept an announced block requires 1102 MS (the block interval time is 3000MS), this is still very fast compared to other blockchains because they usually need 3 to 6 later block confirms to finally accept a block (in our experiment setup, 9000 MS to 18,000 MS). Since, a miner usually sends four Shares per round of the game, it is likely that the peak periods of sending Shares within every round of game, in our experiment, are around 750 MS (3000/4 MS), 1500 MS, 2250 MS and 3000 MS. The average confirm is around 1000 MS which means, which block is the final accept block usually can be figured out in the first peak period of sending Shares of its next block height. Thus, regardless of the exact number of block interval time, the time required for finally accept a block after announced is a rather steady number – around one-fourth of the block interval time. Thus, MWPoW has a dominant advantage in immediate confirms.

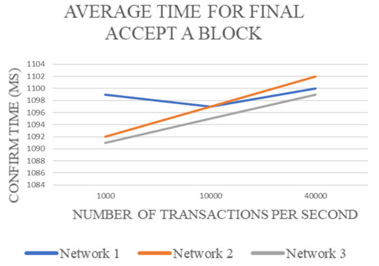


Fig.16 Average time for final accept a block

4.3 Hash difficulty and distribution of funding

To evaluate the hash difficulty and the reward distribution comparison between different blockchains with the same block interval, we run the Nakamoto blockchain with the basic network defined above and the bandwidth scheme of 1Mbytes. The block interval is set to be 5 minutes, the transactions per second are 5, the hash rate for every node is 5 MH/S, the mining reward is set to be fixed 50 coins per block. The accept difficulty is changed every time a new block accepted. The MWPoW used in this experiment is using the same setting as Nakamoto blockchain. We randomly assign CPU frequency to the nodes as to simulate nodes with different calculation ability. Figure 17 shows the change of difficulties of Nakamoto blockchain and MWPoW which are logged by 50 in the period of the experiment from the first minute to 300 minutes since start. Figure 18 suggests that the difficulty of MWPoW is around 1500 times more than Nakamoto blockchain difficulty given the same network situation and same calculation ability per node between two systems. Figure 19 presents the distribution of funding after the experiment ran for 1000 minutes. Most miners in Nakamoto blockchain did not receive rewards, only the minority miners received a tremendous amount of rewards while miners in MWPoW are all received compensations, and the funding is mostly distributed among them.

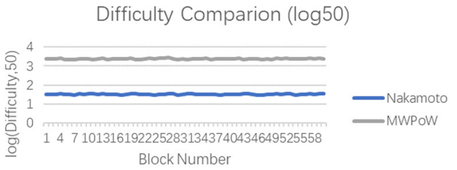


Fig.17 Difficulty Comparison

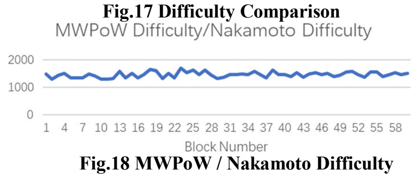


Fig.18 MWPoW / Nakamoto Difficulty

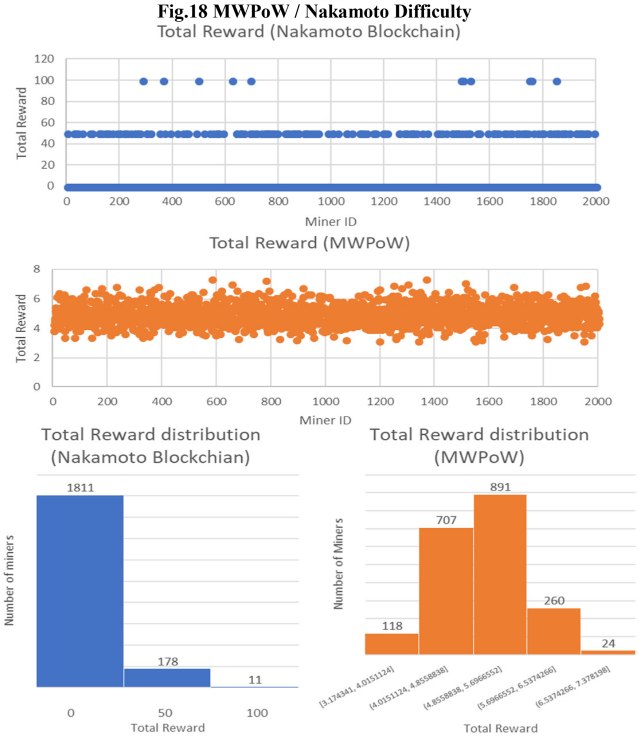


Fig.19 Total Reward distribution

5. Concern

5.1 Hash Difficulty

As the experiment in Section 4 shows, the hash difficulty of MWPoW is much higher than Nakamoto blockchain of the same miner scale. For many, they may think MWPoW wasted lots of energy only to reach consensus but not to do other things. However, as the higher the difficulty of block is, the more difficult it is for attackers to attempt tampering. A good reason for Bitcoin to keep the block interval as 10 minutes but not a smaller number is that would lower the difficulty for attackers to create fake blocks, and nodes may need to wait for more block confirms before final accept a block. Since MWPoW largely higher the difficulty, it is possible and practical to shorter the block interval without afraid the security is damaged.

In terms of wasted energy, the traditional blockchains only show the energy used to first create a block, however it does not show the energy used for creating blocks which failed the game. A block in MWPoW is created by the combined power of around one-third of the valid miners, thus, the energy consumed is not actually higher but, in fact, instead of wasted in vain, the energy is converted to the difficulty of block. Thus, MWPoW can be seen as energy-friendly compared to Nakamoto blockchain.

In addition, the selfish mining [6] can be largely prevented, because the attackers not only need to have the largest calculation to first generate a block by themselves, they need one third of the sum of all the registered power.

5.2 The number of Group

The reason for dividing miners into three groups is that would incent the miners to check the eligibility of the blocks. If there is only one group, all valid miners will receive compensation. Thus, when there comes a block that fulfils the entrance difficulty, miners may start mining directly without checking the eligibility of the transactions embedded in it. When miners are divided into groups, only the miners of the winner group receive compensation, and miners of other groups must abandon their current block and accept the winner block. In which scenario, other groups must check the eligibility of the winner block before quitting their own blocks. The reason for the number 'three' is to prevent the hard-fork, or the delay of final accept a block when miners are choosing branch since abandoning a block is also giving up the remuneration for one round of mining game. If there are two blocks which were announced by two groups that are well-matched in strength at an approximately the same time, and the majority of the miners don't change their branch, the time for final accepts a block may be delayed or a hard-fork may occur. If there are three groups, there is no group taken the approximately half of the power; this guaranteed the smoothness indebtedness of branch choosing and block verification.

6. Related work

P2Pool [8] is an existing decentralized mining pool of Bitcoin. Beside Bitcoin block-chain, P2Pool nodes have a similar cryptographic chain of data representing the value which is called the Sharechain. Shares that are written into Sharechain are the same (cryptographically speaking) as blocks in the Bitcoin blockchain, except that they have a lower difficulty target. When a share of at least minimum p2pool difficulty is found, the node will broadcast it into the P2P network to all the other nodes. If most nodes accepted the share, it becomes confirmed in the Sharechain. A P2Pool share that also meets the Bitcoin difficulty is also broadcasted to Bitcoin peers, and confirmed by the Bitcoin network and becomes a block. The length of Sharechain is 8640 Shares maximum, and it is expected to generate a share every 30 Second. This is known as a Pay Per Last N Shares (PPLNS) payout system, the N in PPLNS is 8,640, which means each of the last 8,640 shares in P2Pool is paid each time a Bitcoin block is found. A share in the P2Pool Sharechain can be expected to last about 3 days (8,640 shares * 30 seconds = 3 days) [8]. Confirmed shares are paid when a block is found if they remain in the Sharechain. It is said that the payouts in P2Pool are made directly from the block generation transaction immediately. However, newly generated coins in Sharechain require 100 confirmations before they can be spent, so P2Pool payouts cannot be spent for about 16.5 hours. A miner in P2Pool is competing with other P2Pool miners for a portion of the 8,640 active shares, the more shares owned in Sharechain the more reward can be given when a block is found.

However, as it is requiring a long pending period before payout the funding, it is inconvenience for miners to conduct immediate transactions in this platform. And since this protocol works as a virtual entity in another blockchain system, the general reward expectation for participants are still considerably low, if there are many participants in the other blockchain system.

7. Conclusion and Future work

In this article, we discussed a new consensus reaching protocol—MWPoW, a blockchain protocol that can achieve consensus among thousands of nodes in a second. In addition to that, one-third of valid miners will be rewarded with remuneration; this helped the distribution of currency. Because the use of BF and IBLT, factors that slow the data propagation and transaction confirm shifted from the size of the block to the number of transactions, NJ and Share sent to the network between every block generation. In the meantime, we admit that in the model of MWPoW, when miners' bandwidth is lower than average, it is easy for them to be expelled from the game, thus, though it allows vast devices to join in the mining game and enabled the immediate confirm but the disadvantage of bandwidth caused the new unequal of the game. To solve the uneven problem ultimately, researches in the future might focus on the design of MWPoW in multi-chain structures, for example, to set an upper and bottom limit of the bandwidth of a chain, split the chains when excelled the upper limit and combine when lower than the bottom border. Thus, the structure can include more transactions faster, and the burden of individual miners can be reduced.

References

1. Karl J. O'Dwyer and David Malone. Bitcoin mining and its energy footprint. In Proceedings of the 2014 IET Irish Signals & Systems Conference, 2014.
2. Burton H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. Commun. ACM, 13(7):422-426, July 1970.
3. Matt Corallo, Bip152:Compact block relay, https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki, April 2016.
4. Decker C, Wattenhofer R. A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels[M]// Stabilization, Safety, and Security of Distributed Systems. Springer International Publishing, 2015:3-18.
5. M.T. Goodrich and M.Mitzenmacher. Invertible bloom lookup tables. In Conf. on Comm, Control, and Computing, pages 792-799, Sept 2011.
6. Eyal I, Sirer E G. Majority Is Not Enough: Bitcoin Mining Is Vulnerable[M]// Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2014.
7. Coindesk.com. Merge-Mining. https://www.coindesk.com/charlie-lee-proposes-merged-mining-litecoin-dogecoin/, Feb 2018.
8. P2Pool. Peer to Peer Mining Pool. http://www.p2pool.org/, Feb 2018.
9. Peter Tschipeer. BUIP010 Xtreme Thinblocks. https://bitco.in/forum/threads/buip010-passed-xtreme-thinblocks.774/, Jan 2016.
10. Sina tech. Alipay per Second. http://www.mpay-pass.com.cn/news/201511/11092649.html, Feb 2018.
11. Bitcoin.it. Block Difficulty. https://en.bitcoin.it/wiki/Difficulty/, Feb 2018.
12. Ozisik A P, Andresen G, Bissias G, et al. Graphene: A New Protocol for Block Propagation Using Set Reconciliation[M]//Data Privacy Management, Cryptocurrencies and Blockchain Technology. Springer, Cham, 2017: 420-428.
13. David Eppstein, Michael T.Goodrich, Frank Uyeda, and George Varghese. What's the Difference? Efficient Set Reconciliation Without Prior Context. In ACM SIGCOMM,2011