A Practical Scalable Blockchain Protocol

Hao Xu

omegasuite.org

haohxu@yahoo.com

**ABSTRACT**

In this paper, we describe a blockchain consensus protocol based on dual-chain of blocks data structure, one chain to record mining rights and another to record transactions. The network timestamps mining rights by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. Transaction blocks are signed by miners recorded in mining right blocks evidencing the right to produce in the same order as they appear in the mining right chain. Each recorded miner may produce multiple transaction blocks up to a limit set by the network, which may be arbitrarily high. Therefore for the time spent to produce one proof-of-work for a mining right block, multiple transaction blocks in the future, instead of one at present, may be produced, thereby improving the network’s ability to process transactions and yet maintain the same, or even better, level of security and freedom of participation..

**CCS Concepts**

**• Networks ➝ Network protocols ➝Application layer protocols ➝ Peer-to-peer protocols • Networks ➝ Network protocols ➝ Network protocol design**

**Keywords**

Blockchain; protocol; transaction; mining right; scalability; throughput; latency.

# INTRODUCTION

Satoshi Nakamoto [1] proposed an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. The system is a peer-to-peer network where nodes can leave and rejoin at will. The network timestamps blocks of transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of computing power. As long as a majority of computing power is controlled by nodes that are not cooperating to attack the network, they will generate the longest chain and outpace attackers. Thus transactions are computationally impractical to reverse and the system would protect sellers from fraud.

Tx Block

Tx Block

Tx Block

MR Block

Best Hash

MR Block

Best Hash

MR Block

Best Hash

**…………**

Because each block of transactions must be accompanied by a proof-of-work, which requires certain amount of time to produce, such systems have an inherently low transaction throughput and long latency. This prevents the systems from being used in most potential applications where an adequate transaction throughput and latency is a minimal requirement. The blockchain scalability trilemma has been informally conjectured. [3] The conjunction is related to scalability, security and decentralization, and suggests that any improvement in one of these aspects would negatively impact on at least one of the other twos.

What needed is not only a secure, decentralized, cryptographic electronic transaction processing system, but also a system that scales gracefully with work load and having latency acceptable to most applications. In this paper, we propose a consensus algorithm using dual-chain data structure. One chain consists of blocks recording mining rights (MR blocks) and the other blocks recording transactions (transaction blocks). Nodes competing for the right to generate transaction blocks by generating mining right blocks with a proof of work required by network and appending it to the MR chain. Winners take turns to generate transaction blocks in the same order as recorded in the MR chain. Transaction blocks are signed by miners as proof of their authenticity. Each winner may produce multiple transaction blocks up to a limit set by the network, which may be arbitrarily high. Therefore for the time spent to produce one proof-of-work for a mining right block, multiple transaction blocks in the future, instead of one at present, may be produced, thereby improving the network’s ability to process work load and yet maintain the same level of security and decentralization. In case a node fails to produce transaction blocks in her round, any node may generate a transaction block with proof of work, which also causes a turn in mining right.

# Mining Right Blocks

A mining right block or MR block contains the hash of the block immediately preceding it, hash of a transaction block, nonce, and a public key. The transaction block referenced in a MR block is called the best block of the MR block as it will be shown later in this paper that miners tend to use hash of the best transaction block known at the time for this field. A MR block is invalid unless its hash is less than a target value required by the network. A mining right block is also invalid unless its best block is a block in the transaction chain following (including) the best block of the preceding mining right block as shown in Fid. 1.

**Figure 1. Relationship between mining right blocks and transaction blocks**

Nodes use a proof-of-work system similar to Adam Back's Hashcash [2] to produce and validate MR blocks. As explained in [1], proof-of-work system solves the problem of determining representation in majority decision making and provides security to the system. If a majority of computing power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. The majority decision is represented by the longest mining right chain, which has the greatest proof-of-work effort invested in it. By referencing a transaction block in a mining right block, the majority decision is not only a decision about who obtains the right to produce transaction blocks in the future, it is also a decision about transactions the network have accepted. Therefore, the network provides the same level security for transactions in the blocks up to the best block as it does for the mining right block.

# Transaction Blocks

**Tx Block**

**MR Block**

**Best Hash**

**MR Block**

**Best Hash**

**MR Block**

**Best Hash**

…………

**Finger**

**B**est block

**Waiting List**

Figu**re 2. Waiting List**

A transaction block is either signed by a miner or accompanied by a proof of work. A transaction block contains the hash of the transaction block immediately preceding it, a number of transactions, nonce, and optionally signature of a miner. Mining awards are also given in transaction blocks. A transaction block containing miner’s signature is called signed block. Otherwise, it is accompanied by a proof-of-work and is called POW block. We set up the nonce for dual purpose. A POW block would have a positive nonce value. And a signed block would have a negative nonce value.

Nodes take turns to produce signed blocks according to the order of blocks in MR chain,. It helps to imagine that there is a finger pointing to a MR block. A signed block is valid if the miner’s signature matches the public key in the MR block pointed by the finger. There is a limit set by the network on the number of transaction blocks a node may produce in her round. A negative nonce value records the sequence number of the signed block in current round. Thus the first block generated by a miner in her round would have -1 for nonce, next block -2, …, –(limit - 1). The nonce of the last block in this round would be the negative height of the next MR block instead of -limit. Thus every node can easily determine whose turn it is to produce transaction blocks without scanning the entire chain. When this last block has been added to the transaction chain, the imaginary finger moves to the next MR block.

A POW block is valid if its hash is less than a target value the network requires. Because signed blocks can be produced much faster than POW blocks, miners would not try to produce POW blocks unless they belive the node of current round is faulty and unable to produce transaction blocks based on the fact that they have not received any block produced by the node for somrtime. When a POW block is added to the transaction chain, the imaginary finger also moves to the next MR block.

# Waiting List

Obviously, the network needs to produce MR blocks at a faster pace than nodes take turns. Otherwise, it would quickly run out of miners available to produce signed blocks. On the other hand, if the system keeps producing MR blocks at a faster pace, eventually miners would find they need to wait a long time between winning the bid for mining right and actually producing signed blocks (and earn awards). This situation is undesirable since a miner in such situation may decide to leave the network until the time the miner expects to produce signed blocks. To solve the problem a waiting list control mechanism is introduced.

Waiting list of a MR block is a section of mining right chain between the MR block pointed to by the finger for the best block of the MR block and the MR block itself as shown in Fig. 2.

Like [1], a base proof-of-work difficulty target (Base Target) for MR blocks is determined by a moving average targeting an average number of MR blocks per hour. A MR block is required not only to meet the Base Target, it must also to meet increased difficulty requirement when its waiting list is longer than a desirable length set by the network, e.g. 40. Specifically, target difficulty doubles for every MR block in the waiting list exceeding the desired length. With increase of waiting list length beyond the desirable level, the work required producing a MR block increases exponentially and very soon it becomes practically impossible and/or economically infeasible to produce more MR block. Miner would wait for the transaction chain to catch up instead of keep mining on the MR chain. Therefore the length of waiting list is controlled around a desired level. This mechanism not only controls length of waiting list, it also incentivize miners keeping transaction chain up to date, because by using the hash of the latest transaction block as best hash in MR block, it helps to reduce length of waiting list and thus difficulty to produce a MR block. And the system becomes healthier overall when every node does its best to keep up-to-date transaction chain.

# bond and forfeiture

An authorized miner may try to attack the network by signing two transaction blocks at the same height. Similar attack could happen in a network according to [4]. The attackers, however, will act very differently in the two kinds of networks after double signing. In [4], the attacker will secretly mine on a side chain and publish the secret chain when delivery is received and the secret chain is longer than the main chain. Seller may counter this type attack by withholding delivery until it becomes very unlikely that the attacker could have a longer secret chain. Here, because existence of waiting list, it he tries to attack the same way as the attacker in [4] does, he will need to mine on the MR chain not from end of chain, but from his currently position in the MR chain, where there are already a number of (e.g., 40) MR blocks behind him in the chain. If it is very difficulty to overthrow a 7-block chain by secret mining, think about 47.

However, the attacker might try to collide with miners behind him in the MR chain. They will produce signed blocks on the published chain. When seller has delivered, they switch to the secret chain. Seller still could counter the attack by withholding delivery. If less than 25% miners are dishonest [6], the chance an attacker finds the next N miners are willing to collaborate is less than 0.25N. It is very easy for the seller to pick a waiting time that she feels comfortable with.

Further, we require miner to include in the MR block as bond one of his unspent transaction output (UTXO) worth an amount B, which is adjusted periodically to average total block transaction amount. If that UTXO has been spent before the miner’s turn, he will be skipped and the next miner gets the right to mine. Otherwise, once a miner produces a valid signed block, the UTXO will be locked until certain period of time calculated at the end of his turn according to a formula described below has passed. If the miner double signs during this period, he forfeit his bond.

Lock time formula consists of two parts: base lock time and value based lock time. Base lock time is a constant in number of turns, e.g. 7 turns. Value based lock time is the sum of total transaction output amounts in the blocks the miner has signed this round divided by B in number of turns. E.g., if a miner has signed 10 blocks in this round, and total transaction output amount in these 10 blocks is 950. If B = 100, value based lock time is 950 / 100 = 10 turns. Thus lock time is 10 + 7 = 17 turns, i.e., the UTXO will be unlocked after the 17-th miner after him has finished his round.

Seller may hold delivery until the total bonds locked since her transaction is recorded exceeds the amount of her transaction.

# Analysis

## Latency and Throughput

The idea of separating mining rights recording and transactions recording is not new. In [4], the block chain consists of two types of blocks: macro blocks recording mining rights and micro blocks recording transactions. The difference is that in [4], nodes begin to generate micro blocks once a macro block is generated, while here, nodes have to wait until her turn. Thus in [4], production of transaction blocks is synchronous with gaining mining right and here they are asynchronous. While both protocols will achieve similar improvements in transaction throughput, they will have different effective latencies. As a result, the protocol presented here will give user better experience and satisfaction and provide better transaction security.

Conventionally, latency is measured as the average time between transaction submission and generation of a block containing the transaction. This, however, is not an accurate metric for measuring user’s perception of latency in blockchain because there are always side chains and a block otherwise valid can be set aside because its branch turns out to be shorter than another branch. We measure user’s perception by effective latency which is a function of confidence in transaction finality: *f(c)*, i.e. how much time a user has to wait in average since submission of a transaction to have a confidence level of *c* in that the transaction will be final.

In Bitcoin network, about one in 60 blocks generated by all nodes ends up in side chains [5], thus a user would have 1-1/60 confidence in that the transaction will be final with first confirmation. Therefore, effective latency *f(59/60)* is 5 minutes for Bitcoin network, half of the average time to produce a block. Generally, for a transaction with n confirmations, confidence is approximately c = 1 – (1/60)n, *f(c)* = 5 + 10 (n-1) minutes.

In [4], side chains could not be eliminated even if all nodes are honest and function normally. Honest nodes may produce valid macro blocks at the same height and generate two chains of micro blocks. Therefore, it would have an effective latency function very close to Bitcoin’s.

Here, barring a successful attack on the network and node failure, there would be no side transaction chain at all. There will still be side chains in MR chains. However, because the network would maintain waiting list long enough, conflicts between side chains would have been resolved long before a node may produce signed blocks. If length of waiting list is 40, user would have 1 – (1/60)40 confidence in that a signed block generated by the node is final without further confirmation. And effective latency *f(1 – (1/60)40)* = 5 seconds, if average time to produce a signed block is 10 seconds.

For a seller willing to deliver when she has 99% confidence in payment, which isn’t too much to ask, if she receives payment in a network according to [4], she would wait 15 minutes before delivering. If she receives payment in a network according to the present protocol, she will deliver within seconds of transaction submission. This makes the protocol suitable for retail applications.

## Security

It is more difficult to attack a network according to the present protocol by secret mining than to attack Bitcoin network or a network according to [4]. Essentially, a secret miner faces the same level of difficulty when he attacks Bitcoin network or a network according to [4]. He starts secret mining immediately after a transaction is recorded and publishes his secret chain when the seller believes that first transaction is finalized and delivers. To succeed, he must overcome the longest chain of blocks produced by all other honest miners during this period. If that is difficult, it is virtually impossible to launch a secret mining attack against a network according the protocol presented here. Because when his first transaction is recorded, there is already a long waiting list in MR chain. He must replace all the MR blocks in the waiting list with his MR blocks and then the new MR blocks other honest nodes produce while he is attacking.

Alternatively, he may try to bribe and collide with the miners after him in line. To counter this type attack, a bond/forfeit protocol is installed. A seller of a large value item could delay delivery until the total amount of bonds provided by the miners who have produced signed blocks since her transaction was recorded exceeds the value of her item. Thus, it becomes unprofitable to attack.

# Results

A block chain network based on a variation of the protocol presented here has been deployed and is accessible at http://omegasuite.org. The network requires two votes by signature of three miners indicated by the three MR blocks preceding the block’s finger and has a minimum block time interval policy. The network steadily generates one block every 8- 10 seconds and maintains a waiting list around 40 blocks in length. We also implemented and tested the protocol in a test environment with 16-way parallelism for signature verification. The test result shows that the network is capable of processing transaction at 1623 TPS..

# Conclusion

We have proposed a block chain consensus protocol for processing electronic transactions effectively. To solve the low throughput and long latency problem inherent in today’s block chain technology, we proposed to separate the right to mine from the act of mining by having two separate chains, each comprise of blocks stringed by a chain of hashes. A node will obtain the right to mine by winning a competition among miners for solving a proof-of-work puzzle. Winner obtains the right to mine a number of transaction blocks in the future. The number of transaction blocks a winner is allowed to produce could be set by system arbitrarily to achieve desired performance. Transaction blocks could be signed by an authorized miner without a proof-of-work. Thus they could be generated quickly and would result in high transaction processing rate and law effective latency. The network is more secure due to the fact that the network maintains a waiting list that is virtually impossible to attack.

# REFERENCES

1. Satoshi Nakamoto 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System*. <https://bitcoin.org/bitcoin.pdf>.
2. A. Back 2002. *Hashcash - a denial of service counter-measure*, <http://www.hashcash.org/papers/hashcash.pdf>
3. H. Halpin 2008. *Deconstructing the Decentralization Trilemma*, arXiv:2008.08014v1
4. Ittay Eyal, Adem Efe Gencer, Emin G ̈un Sirer, Robbert van Renesse, 2015. *Bitcoin-NG: A Scalable Blockchain Protocol*, rXiv:1510.02037v2 [cs.CR] 11 Nov 2015.
5. C. Decker, R. Wattenhofer 2013. *Information propagation in the Bitcoin network*, In IEEE P2P (Trento, Italy, 2013)
6. Eyal, I., and Sirer, E. G. Majority is not enough: Bitcoin mining is vulnerable. In Financial Cryptography and Data Security (Barbados, 2014)
7. H. Xu 2020. Patent application 202010396756.8, China.