Xcoin: A Scalable and Permission-less Blockchain Platform

*Yangrui Guo, Christian Mate Jr*

**Abstract.** Present consensus methods, including Proof-of-Work, achieve a prohibitive transaction throughput for large-scale applications, including the mainstream exchange of the digital currencies that secure them. Additionally, transactions cannot be confirmed until a new block has been mined. We present Xcoin, wherein a dynamic group of replica nodes acts as validator committee within a Byzantine fault tolerance-based blockchain system. The system adopts proof-of-work instead of certificate authority to allow its open participation. Leader election and transaction validation are decoupled into two separate blockchains. Transactions are permanently recorded once verified by more than two-thirds of majority of members within the validator committee.

# Introduction

Throughout the course of human history, the evolution of currencies has never ceased.

From shells, and precious metals, and to the soon-to-be mainstream digital currencies, humans have always been adopting more convenience and faster completion of exchange for commodities. Modern currencies are not associated with the value of physical commodities such as gold or silver, but de facto fiat currencies backed by government’s credit. Fiat currencies possess no intrinsic value; neither do they have the obligation to be converted to gold. Because the right to issue currencies is controlled by the government, fiat currency can be infinitely printed, thus cause inflation. In the event of financial crisis, war or serious government deficiencies, the credit of fiat currencies shall face serious challenges. Moreover, fiat currencies face severe limitations in regard to exchange rate and clearing period when it comes to international transfer. After the global financial crisis of 2008, regulators across the world began implement quantitative easing policy, which resulted sharp drop in value of fiat currencies issued by them. In 2009, Bitcoin was born amidst of this global chaos. Again in 2013, Cyprus crisis has shown how fragile is the centralized financial system dominated by the government and banks. Bitcoin gained the world’s attention and achieved relative success because it does not rely on any centralized issuing authority. Bitcoin’s underlying technology, the blockchain, has begun to profoundly influence every aspect of the Internet and the society. Many enthusiasts believe it will bring the next industrial revolution, and have begun exploring more use cases other than financial transactions, such as contracts, securities, identity verification and logistics. However, it has been over eight years since the genesis of Bitcoin, and the deficiencies of public current public blockchains are becoming more prohibitive to their adoption each day. As the technology advances, Bitcoin will also be succeeded, just like all other currencies, by subsequent technological innovation. In this whitepaper, we present a new blockchain which improves upon the current disadvantages of public blockchains.

# Cryptocurrencies

Cryptocurrencies are a series of secure, immutable records of transactions. In real life, when a person wishes to transfer their money to another party, they may write a check and sign it with his signature. Cryptocurrencies are based on the same principle: each asymmetric public key represents an address, and accordingly it comes with a private key. A public key can be derived from the private key, but not the other way around. Everyone can see the public key, but only the address owner can access his own private key. If the owner wants to spend the balance in his address, he must sign a transaction with his private key so the transaction becomes valid. Cryptocurrencies have no physical form and only exist within a computer network. Xcoin is a cryptocurrency based on blockchain technology. Xcoins act as the incentive to secure the blockchain they are transacted over.

# Blockchain

Because there is negligible work involved in signing a transaction, a cryptocurrency can be easily duplicated. To prevent the same coin from being spent multiple times, the so-called 'double spending problem', cryptocurrencies introduced the concept of blockchain. A blockchain is a cryptography enhanced immutable database format. A block can be deemed as a piece of paper with a few of transactions written on it. Each block then contains a header, which includes a hash of its previous block along with current block’s timestamp, Merkle root and other information. Therefore, with the exception of the genesis block, every block is 'chained' after its previous block, thus forming a block-chain. A public blockchain is a database of every transaction that has occurred since its deployment, and anyone may read its data. So long as a group of nodes maintain a distributed network to broadcast, validate, and store transactions, it will grow indefinitely. The number each block corresponds to is called block height. In terms of cryptocurrency, due to the possibility of tampering during data transmission and storage process, we must adopt a tamper-proof to protect our data. To eliminate the dependence of a centralized issuing authority, blockchain utilizes a peer-to-peer network to handle transactions. Each node contains an identical copy of the blockchain for verifying whether the data has been altered. The hash algorithm of a blockchain is designed that even the smallest change to a record can cause all records after it become drastically different. To modify a block at a certain point in the past, the entire blockchain after it must be recalculated. Hence, it is practically impossible.

# Consensus

Consensus is the most crucial part of decentralized peer-to-peer network, and is imperative for a permission-less blockchain. Due to objective factors, latency, downtime and malicious attack are prevalent among distributed systems. When two nodes have inconsistent block records, a fork occurs. This can cause the problem of double-spending: if node A contains a transaction which is not found on node B, then who is to make the determination of whether or not the transaction was valid? To address this, a blockchain requires every participant follow the exactly same rules in order to keep their state stay synchronized. Currently, common blockchain consensus mechanisms include: Proof-of-Work, Proof-of-Stake and Byzantine Fault Tolerance.

Proof-of-work was first proposed by Hashcash as an anti-spamming technique, and later saw adoption in Bitcoin and Ethereum. It is the most classic and prevalent consensus mechanism within the current blockchain landscape. The principle of Proof-of-work requires a hard-to-obtain, while easy-to-verify hash value to be located by a node before it is accepted by its peers. This method is designed in order to prevent the 'Sybil attack', wherein an attacker can produce millions of fake nodes to replace honest ones. Specifically, a proof-of-work is obtained through repeated calculations of the block header until a hash less than a certain difficulty is found, i.e., the “mining” process. All nodes consider the blockchain containing the most recent proof-of-work to be the “real” chain. Therefore, to control the blockchain, an attacker must possess more than 51% total computing power of the entire network. This mechanism is also called permission-less consensus, which allows nodes to join and exit anonymously at any time. Although Proof-of-work is one of the most secure permission-less consensus algorithms to date, its drawbacks will likely hinder mainstream adaptation for high volume applications. Currently, The Bitcoin network generates a block every ten minutes, with a maximum block size of 1MB. This has restricted Bitcoin’s transaction speed to no more than 7 transactions per second. Moreover, because of Bitcoin’s inherent forking problem, a transaction can be considered secure only after six confirmations, which takes a minimum of an hour. This performance is intolerable for modern payment processing. While VISA is capable of processing 2000 transactions per second on average, and may handle over 10000 transactions per second during its peak. The implications of raising the block-size have been debated within the Bitcoin community for several years now. However, successfully doing so would require a significantly large sum of the network to change their software, effectively creating a fork of the Bitcoin blockchain, until the entire network resolves the leader. Two popular and seemingly acceptable solutions are side-chains, and the lightning network, both of which exist off Bitcoin blockchain and inevitably bring centralization and security problems.

Since Proof-of-work also consumes a relatively large amount of energy, there have also been proposals of consensus by the ownership of stakes, or, 'coins'. In this schema, nodes express the acceptance of a transaction by signing a portion of their coins as deposit. If a node commits fraud, its deposit is forfeited as punishment. However, the initial distribution of coins is questionable. Because wealthier nodes have more voting power, and the coin value can be manipulated, this method naturally comes with centralization. Moreover, it is far less secure than Proof-of-work because any stakeholder can initiate a nothing-at-stake attack by signing multiple blockchain histories. The centralized validator nodes also create single-point-of-failure. If nodes collude together or are taken down by hackers or the authority, the entire network goes down as well. Although Proof-of-stake has been proposed as early as 2012 by Peercoin, it has not seen as wide of acceptance as the Proof-of-work mechanism.

The consensus problem may also be generalized as the Byzantine general’s problem, wherein nodes within a system can exhibit any abnormal behavior while the system must continue to function normally. It has been proven that in order to tolerate *f* number of Byzantine nodes, at least 3*f*+1 nodes must exist in the system. Within the peer-to-peer network of the blockchain, nodes can be disconnected, shutdown or broadcast malicious information so it is impossible to predict their behavior. In 1999, researchers at MIT for the first time published the practical Byzantine fault tolerance algorithm, which lowered the complexity of Byzantine algorithm from exponential to polynomial. This algorithm choses one node as primary, and a group of nodes as backups. Consensus is achieved after multiple rounds of voting among nodes. This method is originally designed for internal closed systems, such as flight control system of airplanes and server cluster of data centers. It is not suitable for decentralized network of cryptocurrencies. Famous examples of Byzantine fault tolerance based blockchains are Hyperledger and Stellar. In these examples, mining is abandoned and transaction speeds of up to 100,000tx/s are possible. Similarly, PBFT does not support nodes to join and exit the network at will. When the total number of nodes go beyond 20, the entire network drastically slows down.

# Design goal

Because an attacker can easily create hundreds dishonest nodes at almost no cost, in Proof-of-stake and PBFT-based blockchains, nodes must get authorization prior to entering the network. Therefore, the governance of nodes is tightly held by the developers of the blockchain. A cryptocurrency is effectively worthless if it is not built on top of a trustless mechanism, and becomes no different from a fiat currency if issued by a central authority. We seek to develop a blockchain network that retains both Bitcoin's decentralization and the capability to process thousands of transactions-per-second. Here we present a hybrid consensus mechanism, which serves as the foundation of Xcoin.

The purpose of Bitcoin mining is essentially the process of leader election within its peer-to-peer network, which subsequently makes it more expensive to modify the blockchain. This process, however, is not strongly related with the transaction data, which exists within the block-body. Therefore, the mining process can be separated into two distinct chains: the election chain, which uses Proof-of-work to determine a group of leaders, and the transaction chain, based upon Byzantine fault tolerance. There is no confirmation time for transactions to be processed. The first instance of the decoupling of leader election and transaction processing was first proposed by Bitcoin-NG. However, the initial design was flawed, because the leader node has complete deterministic control, and if it goes down, the network is unable to continue. We present a new consensus mechanism that builds upon concepts presented in Bitcoin-NG, but is more robust, and satisfies the requirements of decentralization.

Next, we draw an analogy between the voting process of United Nations and the consensus mechanism of Xcoin. When United Nations resolves an issue, it launches a vote over a draft of resolution proposed by one of its member countries. Then delegates from all countries start voting on the draft. If the draft receives more than 2/3 approvals of all voters, it becomes a resolution. Five Security Council members have veto power. However, as time passes the power of decision-making will eventually fall into those five council members. In a decentralized cryptocurrency, the centralization of power is the last thing we would like to happen. Our solution is to limit the tenure of voters, and allow all nodes to have an equal opportunity to vote. Moreover, since the Xcoin blockchain is permissionless, and all nodes may join and exit without authorization: we must maintain the integrity of the data. By doing so, if malicious actor produces numerous nodes via a botnet in order to gain more than 2/3 of the voting power, the honest node network still retains leadership. In order to accomplish this, we hereby adopted Proof-of-work in order to prevent a Sybil attack.

## Election chain

For fair competition, Xcoin determines the list of validator committees via the process of mining. Each election block represents an interchange in the validator committee. Before the election process begins, the committee size *n* is first determined, then all competing nodes mine the next block. When a block is mined by a node, it is broadcast to the network and verified by other nodes. Before timeout, the first *n* nodes to find the block hash then become the validator committee for the voting next round. If an attacker attempts to modify a block at height H, he must find all *n* and outrun honest nodes.

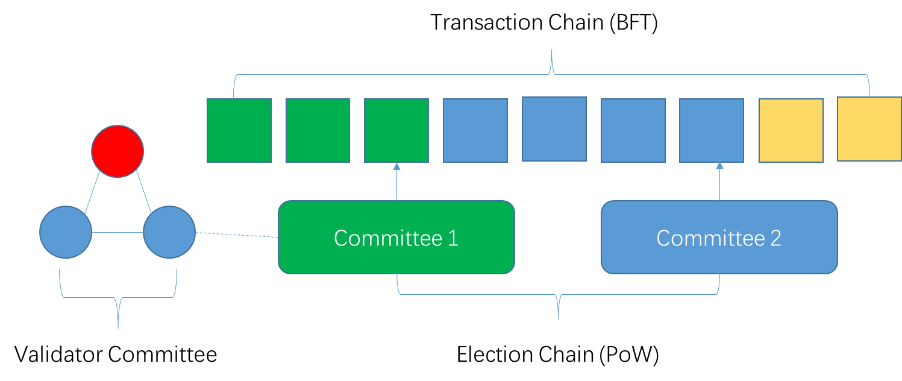
B(*H*) = Hash(M1(*H*) + M2(*H*) + … + M*n*(*H*))

After the mining is finished, the successful competing nodes become validator nodes. The first node that finds the next block automatically becomes the leader for the committee, which is responsible for collecting transactions into transaction block as well as coordinating other committee members. Each transaction block must be approved by at least *n*\*2/3 committee members before being accepted to the transaction chain. After the tenure is ended, incumbent nodes become competing nodes and the validator committee is replaced by the next group of winners. Each honest node receives mining reward after their tenure has ended. Although nodes can freely join and exit the network, they must possess a valid Proof-of-work to do so, effectively preventing a sybil attack.

It is possible for a temporary fork to occur before the committee is determined if more than one node broadcasts simultaneously for the *n*th position. However, this situation may be addressed by a simple deterministic function that picks only one node as final result.

## Transaction chain

Within Bitcoin’s consensus, only one block exists at a certain height, because a transaction cannot be spent more than once. This not only generates several unconfirmed orphaned blocks, but also leads to prolonged transaction time. Under the double chain mechanism, because the consensus of the validator committee is already achieved before voting starts, all transaction blocks are verified instantly without mining and confirmation. The consensus of transaction blocks is achieved via Byzantine fault tolerance. A transaction block includes transaction records, Merkle root, UNIX timestamp, hash of the last block and the current election chain.

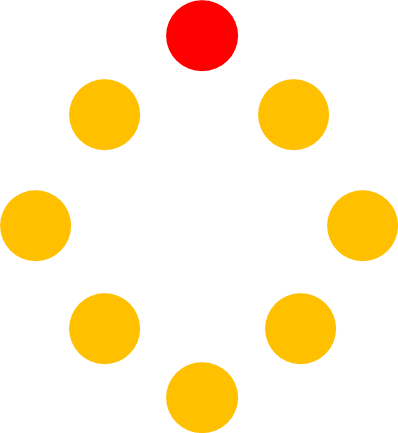


## Voting process

The Xcoin consensus is de facto the process of state machine replication. Each round of validation consists of three phases: pre-prepare, prepare and commit.

### Pre-prepare

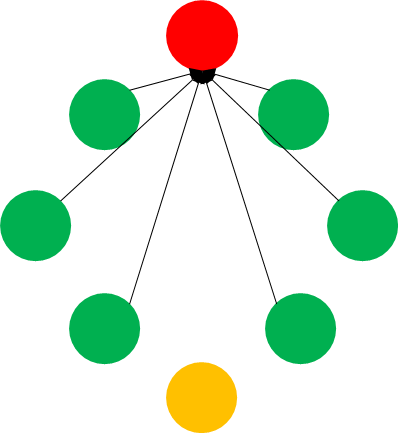
The leader node sends a message to all incumbent nodes and waits for their reply.



### Prepare

Incumbent node validates the message and sends a response to the leader node. If 2/3 majority of a quorum is achieved, then the leader broadcasts announces to proceed to the next phase. During the voting stage, a node can have three possible kinds of responses:

1. The node approves the message.
2. The node refused to approve the message.
3. The node did not respond before timeout.

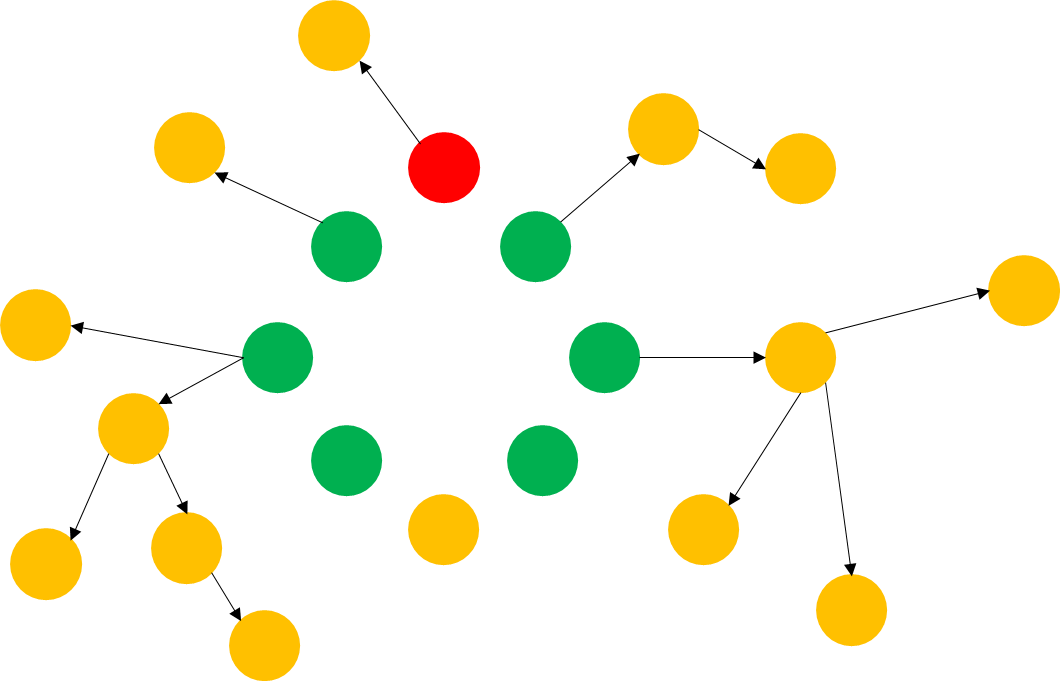


Non-responding nodes are routinely excluded from the validator committee. Therefore, the voting process continues in the event of more than 1/3 validators become unreachable.

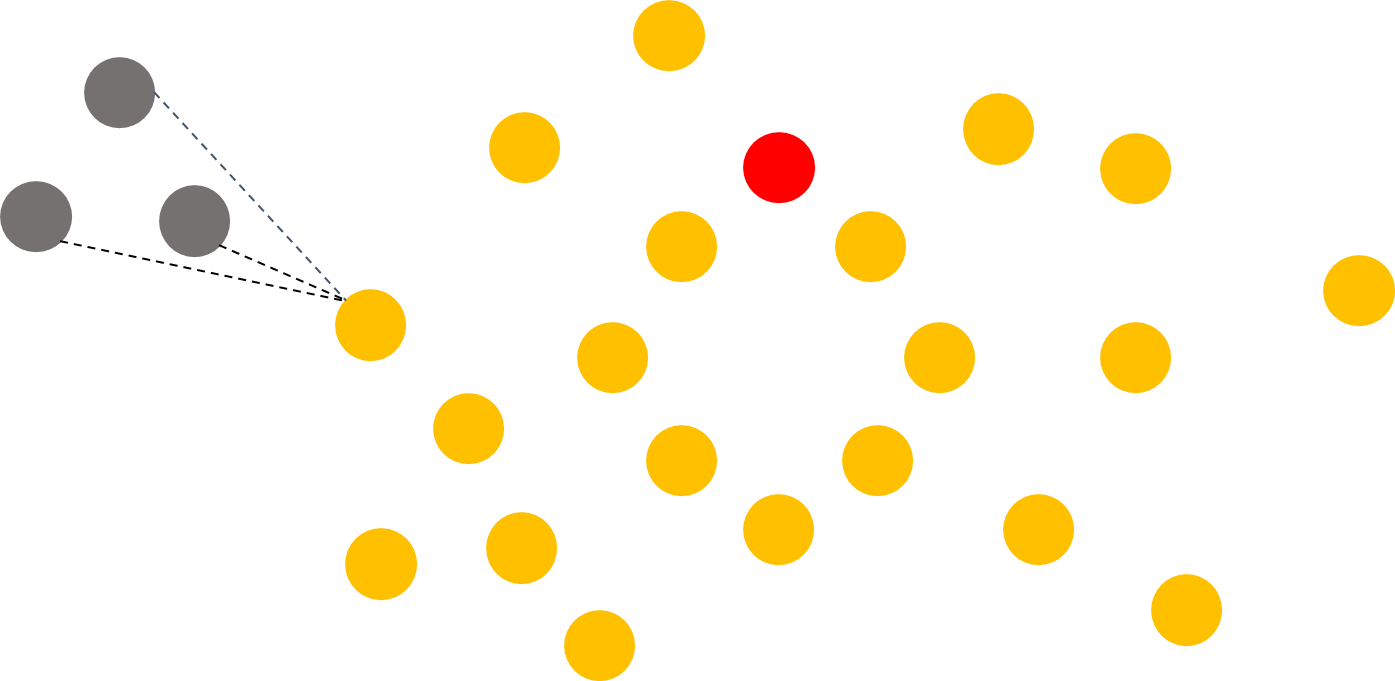
### Commit

If a block receives 2/3 or more approval from incumbent validators, it will be committed to the blockchain and broadcast to all nodes in the network.

If the leader node commits malicious behavior or fails to respond within a set time, other validators can impeach the leader by initiating a view change to replace the leader with the next candidate.



An attacker cannot outvote honest nodes by creating millions of fake nodes to control the 2/3 majority, because all validators must provide proof-of-work before the network accepts them.



## Multi-sig voting

To illustrate the voting process, we draw an analogy between the voting process of United Nations and Xcoin. In the General Assembly, delegates of each country can sign the draft as approval and each other can see who has signed. However, in peer-to-peer system there is no central authority that maintains identities of all nodes. How can the network make sure that all signatures are trusted? A simple approach is for each node to keep a list of all public keys of other nodes, which can be used to verify against their signatures. This means that for total of N nodes, each verification step takes N2 times of check against signatures. When the number of node grows, it could become considerably expensive to verify all signatures. Hence, we introduce a collective signing algorithm. It can produce a collective signature with a group of decentralized nodes so that each node only needs to verify this signature once. The signing process has four steps:

### Announcement

The leader announces a new round of voting, and multicast the announcement to all validator nodes via a spanning tree from top to bottom.

### Commitment

When a node receives the message from the leader, it generates a random key and uses that key to compute a Schnorr signature. Then it combines this signature with its direct child nodes and sends back to the leader.

### Challenge

The leader computes a Schnorr challenge with a hash function, then multicast it along with the message M to all validators.

### Response

With the collective challenge from the last step, all nodes return a combined response from leaf to root.

The leader executes the signing process first time during the pre-prepare phase. However, some nodes may lose contact, and then the leader must execute the signing process again during the commit phase. Two rounds of signing guarantee that the result comes from all responsive nodes.

Tree multicast is flawed in itself; its stability is crucially depending on the leader node. We made improvements of this protocol to ensure that in the events of leader failure, the rest of nodes can continue to operate without causing errors.

# Smart contract

To support more advanced and flexible transactions, Xcoin comes with a stack-based scripting system, also known as a smart contract. The smart contract runs inside an Xcoin virtual machine. Smart contracts can be programmed to create more complex decentralized apps, such as deferred payment, advanced access management and user defined digital assets. The high throughput of Xcoin is capable of satisfying large-scale requirements of enterprise applications. The syntax of the scripting system will be similar to Bitcoin’s, which can perform basic mathematical operations and logic expressions.

## Use cases

Alongside value transfer, an Xcoin smart contract can support storage for various formats of data. The distributed storage of Xcoin blockchain provides better stability and is ideal for anyone seeking high availability and security of their information Below is an incomplete list of use case scenarios of Xcoin.

### Finance

Stock, bonds and other financial instruments can be implemented using Xcoin blockchain. Comparing with traditional exchange centered system, blockchain based solution is more effective and costs less.

### Digital contract

An agreement between two or more parties can be signed with their digital signatures, which renders the agreement immutable in future. An Xcoin smart contract can also implement asset ownership, wills and certifications.

### Messaging

Xcoin’s built-in transaction fee mechanism can effectively prevent spamming. This allows users for more valuable and secure information exchange.

### Voting

Xcoin can permanently store voting records for organizations and institutions, such as shareholder voting, governmental bidding and ballots. Voting result can be executed immediately, or deferred until a time in future.

### Notarization

Traditional notarization relies on a trusted third party agency. The Xcoin blockchain can directly verify information while improving efficiency and security compared to traditional methods.

### Secure data storage

Xcoin’s dynamic node election mechanism ensures that all nodes are replaced after a certain period, therefore minimizes the possibility of single point of failure. This feature makes it ideal to store high value critical data.

# Cryptographic algorithms

## Mining

The process of mining is defined as nodes competing by their speed of hashing the block header, until a value less than the current difficulty *d* is found by one of the nodes. This difficulty is adjusted according to total computing power in the network in order to maintain the mining time within a certain range. Because it is virtually impossible to decipher the block header from the hash value, nodes must repeatedly try with a random nonce until the correct hash is discovered. As *d* becomes smaller, the computing power required to find the hash becomes proportionally larger. Although it requires billions of calculations to mine the correct hash, it only takes one calculation to verify whether the hash is correct. The mining algorithm used by Bitcoin, SHA256, is vulnerable to Application Specific Integrated Circuits (ASIC) because they have significant advantages over CPU for mining purpose. The centralized production and employment of ASICs also contributes to centralization within a permission-less system. Xcoin proposes the use of an ASIC-resistant CPU mining algorithm which is bound to device memory.

## Address generation

Xcoin uses Ed25519, an implementation of Schnorr signature algorithm, for its address generation. Ed25519 is faster than ECDSA, and has smaller size, thus making it the ideal signature algorithm for cryptocurrency. A quad-core Intel 2.4G CPU (Xeon E5620) can validate 109,000 of Ed25519 key-pairs per second.

The process of address generation:

1. A Ed25519 key pair is generated
2. A SHA3-256 hash is generated from step 1
3. The RIPEMD-160 of step 2 is calculated
4. The Base58 string is computed from step 3. This is the final unique address

# Transaction

Traditional centralized payment-processing works by maintaining a list, which contains account, balances of all users. A transfer in such system is simply addition of one account and subtraction of the other. All transactions of an account can be easily linked. To protect user privacy and decentralization, Xcoin perceives account balances as unspent transaction outputs, similar to bitcoin. In the transaction chain, each individual transaction consists of one or more inputs and outputs. Excluding the coinbase-output, input transactions are outputs from previous transactions, which designates this specific address as receiver. The input also contains sender’s signature, which can be verified against his public key. The output contains receiver’s address. The balance of an account is the sum of all UTXO associated with it. An UTXO can only be spent once and the remainder will be refunded to the sender as change. Furthermore, client nodes can prune transactions that are too old and only keep UTXO.

## Coinbase transaction

Coinbase is a special variant of a transaction. It does not have a sender address, but is awarded to validator nodes for validating transactions. This reward is determined by the current reward of the network. Unlike Bitcoin, the coinbase transaction of Xcoin does not occur immediately after an election block is mined, but after the tenure of current validator committee has ended. This mechanism effectively ensures the honesty of nodes and suppresses selfish-mining in Bitcoin.

## Transaction fees

Because nodes consume resources such as electricity and bandwidth, and computing power in order to validate transactions, it is necessary for the sender of a transaction to include a fee proportionate to the resources required to process it. This mechanism also mitigates spam or 'dust' transactions. The difference between all transaction inputs and output is the transaction fee paid to the nodes.

## Multi-sig

Sometimes it is insufficient to have only one signature for security concerns. Multi-sig requires all or at least two parties to sign the transaction before it can be executed. The Xcoin scripting language will provide native syntax for multi-signature transactions.

## Transaction verification

When nodes begin syncing the blockchain, they first pick the election chain containing the most recent, longest chain of proof-of-work, then verify whether each transaction block associated with the election chain contains the collective signature of the validator committee. The verification of a single transaction is done by the incumbent validators. If one of the following conditions is true, then the transaction is considered invalid:

The signature of the transaction does not match sender’s private key

The total output exceeds total input

The input transaction has already been spent

Nodes can validate the occurrence of a prior transaction by checking the Merkle tree. A Merkle tree is a binary tree by combining the hash of two adjacent transactions, until there is only one hash left. This hash is called the Merkle root. Any change to in the block will cause this value to change. Each transaction block contains the Merkle root of all transactions within. Through merging Merkle tree branches, a node can verify the transaction without downloading the entire blockchain.

# Security

An attacker cannot forge a transaction of another user without obtaining the user’s private key first. In the worst-case scenario, he can only attempt to recall his own transactions by modifying the block history. All nodes must check if a key block contains enough Proof-of-work before accepting it and all transaction blocks associated with it.

If a node becomes aware that it is maliciously excluded by another validator, it can broadcast view change to other validators and have the malicious validator removed from the committee. However, merely corrupting 1/3 validators is insufficient to allow the attacker rewrite the blockchain history. Consequently, a malicious actor also must outvote the rest 2/3 honest validators to get the transaction block passed, and requires over 2/3 total computing power in the network to do so. In the unlikely event the attacker is able to outrun all honest nodes, he can only reverse his own payment in the past, which is unlikely to yield more profit than being honest. The rest of nodes can still verify all transactions if they are willing to.

The validators does not receive their reward immediately, but only after they have validated all transaction blocks designated to them. If during any round, a node goes offline, it will not receive any reward, and thus, nodes are encouraged to continue validating.

# Conclusion

We have proposed a reconfigurable Byzantine fault tolerance based blockchain without permission authority. To allow anonymous participation of everyone while being resistant to Sybil attack, we introduced Proof-of-work as the node election mechanism. The elected nodes will only hold temporary position to verify transactions, thus the network will retain its decentralization. Confirmation time is no longer required as the mining process and transaction verification are decoupled; therefore, transactions are confirmed without waiting. Our system has shown promise to mitigate the scalability problem that permission-less blockchain currently facing. The system can be further developed into different use cases such as permissioned and Proof-of-stake blockchains.

# References

[1] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[2] Marko Vukolić, The Quest for Scalable Blockchain Fabric:Proof-of-Work vs. BFT Replication, Volume 9591 of the series *Lecture Notes in Computer Science* pp 112-125, 2015.

[3] Rafael Pass and Elaine Shi, Hybrid Consensus: Scalable Permissionless Consensus, <https://eprint.iacr.org/2016/917.pdf>, 2016.

[4] I. Eyal, A. E. Gencer, E. G. Sirer and R. V. Renesse. Bitcoin-NG: A Scalable Blockchain Protocol. NSDI, 2016.

[5] Silvio Micali, ALGORAND: The Efficient and Democratic Ledger, https://arxiv.org/abs/1607.01341, 2016.

[6] Jae Kwon, Tendermint: Consensus without mining, http://tendermint.com/docs/

tendermint.pdf, 2014.

[7] Sunny King and Scott Nadal, Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, https://peercoin.net/assets/paper/peercoin-paper.pdf, 2012.

[8] Miguel Castro and Barbara Liskov, Practical Byzantine Fault Tolerance. *3rd USENIX Symposium on Operating Systems Design and Implementation* (OSDI), February 1999.

[9] Ittay Eyal and Emin Gün Sirer, Majority is not enough: Bitcoin mining is vulnerable. *Financial Cryptography and Data Security*, Springer, pp. 436–454, 2014.

[10] David Mazières, The Stellar consensus protocol: A federated model for internet- level consensus, https://www.stellar.org/papers/stellar-consensus-protocol.pdf, November 2015.

[11] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, Bryan Ford, Keeping Authorities “Honest or Bust” with Decentralized Witness Cosigning, *37th IEEE Symposium on Security and Privacy*, May 2016.

[12] Alysson Neves Bessani, João Sousa, and Eduardo Adílio Pelinson Alchieri, State machine replication for the masses with BFT-SMART, *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN 2014, pp. 355-362, 2014.

[13] Gavin Wood, Ethereum: A secure decentralised generalised transaction ledger, http://gavwood.com/paper.pdf, 2015.

[14] BitFury Group, Proof of Stake versus Proof of Work White Paper, http://bitfury .com/content/5-white-papers-r esear ch/pos-vs-po w-1.0.2.pdf, 2015

[15] Nicolas van Saberhagen, CryptoNote v 2.0, <https://cryptonote.org/whitepaper.pdf>, 2013